



TECHNICKÁ UNIVERZITA V LIBERCI
Fakulta mechatroniky a mezipředmětových inženýrských
studií

DIPLOMOVÁ PRÁCE

Internetová podpora předmětu
„Číslicové řízení“

Internet support of the
„Numerical Control“

Prohlášení

Byl jsem seznámen s tím, že na mou diplomovou práci se plně vztahuje zákon č. 121/2000 o právu autorském, zejména §60 (školní dílo).

Beru na vědomí, že TU Liberec má právo na uzavření licenční smlouvy o užití mé DP a prohlašuji, že **s o u h l a s í m** s případným užitím mé diplomové práce (prodej, zapůjčení apod.).

Jsem si vědom toho, že užít své diplomové práce či poskytnout licenci k jejímu využití mohu jen se souhlasem TU Liberec, která má právo ode mne požadovat přiměřený příspěvek na úhradu nákladů, vynaložených univerzitou na vytvoření díla (až do jejich skutečné výše).

Diplomovou práci jsem vypracoval samostatně s použitím uvedené literatury a na základě konzultací s vedoucím diplomové práce a konzultantem.

Datum

Podpis

PODĚKOVÁNÍ

Děkuji panu Doc. Ing. Osvaldu Modrlákovi CSc. za to, že byl vždy ochoten pomoci v odborných otázkách týkajících se diplomové práce a za jeho věcné připomínky.

ANOTACE

Diplomová práce má za úkol vytvořit internetovou podporu k předmětu Číslicové řízení vyučovanému na TU Liberec. Jedná se o skripta k výuce, testy k procvičování teoretických znalostí studentů a programy pro výpočty základních úloh, realizované pomocí Matlab Web Serveru, který umožňuje provádět výpočty v Matlabu, aniž by byl na daném počítači instalován.

ABSTRACT

The diploma thesis has the task to create an internet support for the subject „Numerical Control“. This subject is taught at the Technical University in Liberec. The diploma thesis will contain an university textbook to aid teaching. There will be tests which will enable students to practice their theoretical knowledge and programs for calculating the basic tasks. The programs will be realized with the help of Matlab Web Server. It enables to execute calculations in Matlab, even if the program Matlab is not installed on the computer it self.

Seznam nepoužívanějších výrazů

K výkladu nepoužívanějších termínů byl použit Výkladový slovník výpočetní techniky a komunikací (viz. Literatura [7])

ATA	<i>Advanced Technology Attachment</i> – Oficiální označení pro rozhraní pevných disků, známé spíše jako IDE
cgi-bin	Nejčastější označení adresáře na serveru WWW, ve kterém jsou uloženy programy
DNS	<i>Domain Name Server (System)</i> – Doménový pojmenovávací systém, převádí doménová jména na IP adresy a obráceně
E-mail	Electronic mail – Souhrn programů a systémového vybavení pro síťovou komunikaci mezi uživateli počítačů
Ethernet	Typ počítačové sítě, používaný dnes zejména v oblasti lokálních počítačových sítí
GIF	<i>Graphic Interchange Format</i> – Grafický obrazový formát umožňující ukládat velice úsporné animace a může být průhledný
HTML	<i>HyperText Markup Language</i> – Kódovací jazyk používaný k vytváření formátovaných dokumentů na webu
HTTP	<i>HyperText Transport Protocol</i> – Protokol určený k transportu hypertextových souborů po Internetu
Hypertext	V počítači uložený textový dokument vázaný na speciální aplikaci, která dokáže jeho jednotlivé části zpřístupnit uživateli
Internet	Celosvětová informační a komunikační síť postavená na protokolu TCP/IP
IE	<i>Internet Explorer</i> – Prohlížeč Internetu vytvořený firmou Microsoft
IP	<i>Internet Protocol</i> – Síťový unixový protokol, vycházející z modelu OSI (vrstva 3)
IP address	Adresa počítače v síti používající protokol IP, identifikuje počítač jedinečně v celosvětové síti Internet (čtyři osmibitová čísla)
JAVA	Síťově orientovaný interpretovaný programovací jazyk vyvinutý firmou Sun Microsystems (umožňuje tvorbu appletů)
Mosaic	Jeden z prvních prohlížečů (browserů) WWW
Mozilla	První jméno pro prohlížeč Internetu od firmy Netscape, nahrazeno názvem Navigátor
NSCA	<i>National Center for Supercomputing Applications</i> – Součást University of Illinois, byl zde vytvořen program NSCA Mosaic
OS	<i>Operating System</i> – Podstatné softwarové vybavení počítače, které provádí základní řízení veškerých zdrojů počítače a komunikaci s uživatelem
TCP/IP	<i>Transmission Control Protocol/Internet Protocol</i> – Přenosový a komunikační protokol
UDP	<i>User Datagram Protocol</i> – Standardizované protokoly, které poskytují jednoduché datagramové služby (nespojované)
URL	<i>Universal (Uniform) Resource Locator</i> – Internetový ekvivalent pro adresu
W3C	Organizace definující standardy pro WWW
WWW	<i>World Wide Web</i> – Graficky orientované zpracování informací seskládaných do www stránek

OBSAH

Úvod	6
1. Struktura internetových stránek	7
2. Předmět „Číslicové řízení“	9
3. Instalace	10
3.1 Apache Httpd Server	10
3.2 Matlab Web Server	12
3.3 PHP	13
4. Tvorba internetových stránek	15
4.1 Deklarace formuláře	15
4.2 Prvky formuláře	16
4.3 Nekompatibilita internetových prohlížečů	17
5 Vytvoření úloh pro Matlab Web Server	18
5.1 Vstupní stránka	19
5.2 Výstupní stránka	21
5.3 Zpracování na straně serveru (m-file)	22
5.3.1 Načítání a odesílání dat	22
5.3.2 Simulace úloh	23
5.3.3 Zobrazení průběhů	26
5.3.4 Číslo nebo řetězec znaků	27
5.3.5 Problém	27
6. Toolbox CR	28
6.1 Instalace toolboxu CR	28
6.2 Funkce toolboxu CR	29
6.2.1 Pinitserver.m	29
6.2.2 Pm.m	29
6.2.3 Kraceni.m	30
6.2.4 Zjistiqh.m	34
6.2.5 Carka.m	35
6.2.6 Hvezdicka.m	36
6.2.7 Oriznuti.m	36
7. Realizované úlohy	37
7.1 Funkce pro výpočty do Matlabu	37
7.2 Úlohy v Matlab Web Serveru	38
8. Odkazy na hypertextové dokumenty	40
Závěr	42
Literatura	43
Přílohy	44

Úvod

Tato diplomová práce má za úkol podpořit výuku předmětu „Číslicové řízení“ pomocí internetu. Jejím cílem není úplné nahrazení výuky reálného učitele virtuálním učitelem, ale snaží se pouze o zlepšení vědomostí studentů. Osobní kontakt učitel - žák je při výuce nenahraditelný a studenti dosahují daleko lepších výsledků, než by dosáhli pouze pomocí elektronické výuky.

Ve světě je již několik let velmi rozšířené vyučování pomocí e-learningu. Jedná se o vyučování pomocí interaktivních počítačových programů, distribuovaných například pomocí kompaktních disků nebo lépe pomocí internetu. V případě distribuce pomocí internetu je velká výhoda v možnosti časté aktualizace. Pokud chceme o programu hovořit jako o e-learningové aplikaci, nesmí to být pouze elektronická verze papírové knihy, která se zabývá daným problémem, ale měl by obsahovat i další prvky pro snadnější pochopení a kontrolu nastrádaných vědomostí.

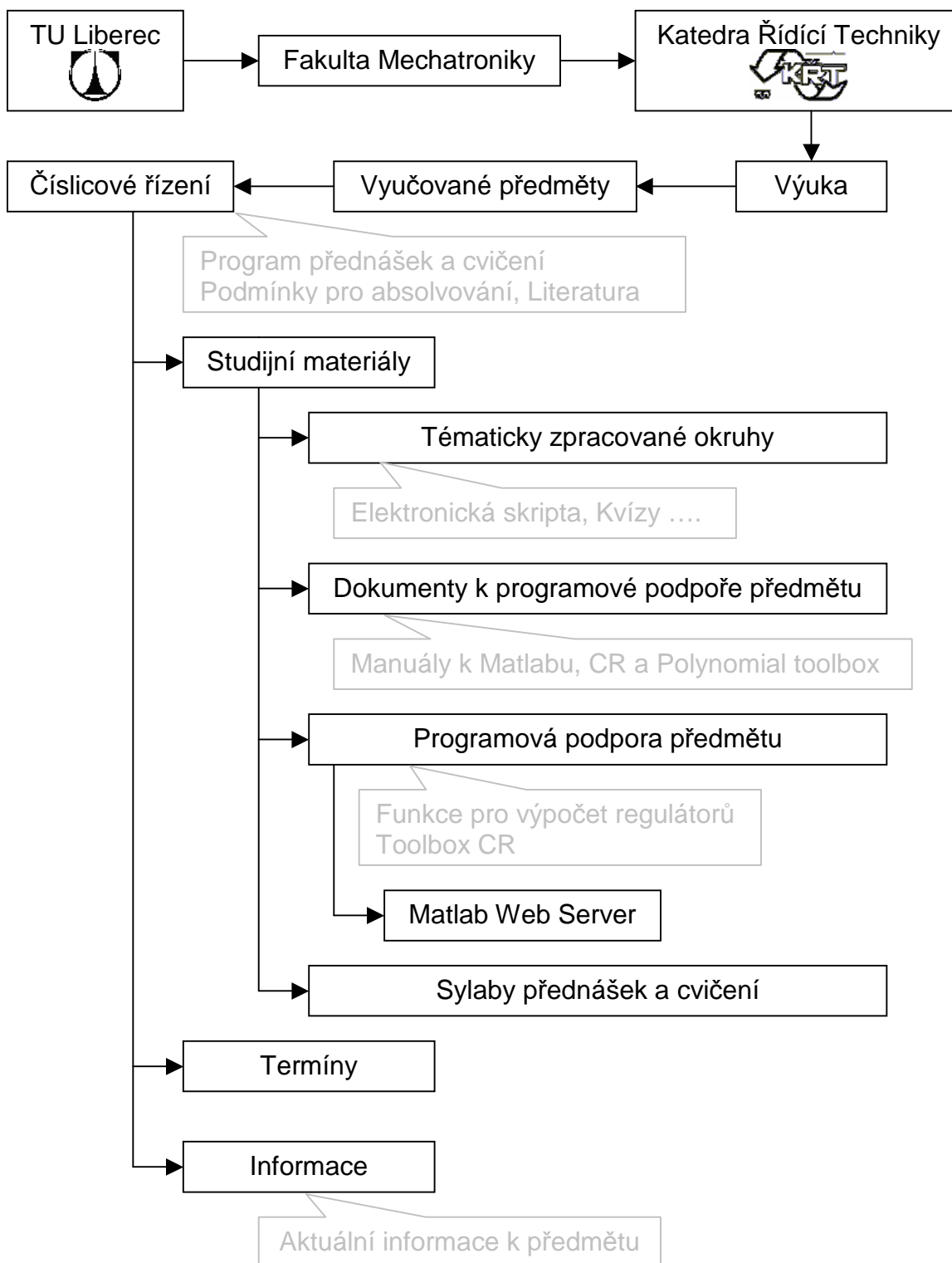
Diplomová práce zpracovává právě takovou e-learningovou aplikaci, která je distribuována pomocí internetu a je volně přístupná.

Základem každé výuky jsou teoretické znalosti, které jsou v tomto případě reprezentovány skripty v elektronické podobě (pdf formát). Jak dobře zvládá student danou problematiku, si prověří pomocí kvízů, kde odpovídá na základní otázky pokrývající celý teoretický základ. Pro snadnější pochopení problému jsou pomocí Matlab Web Serveru realizovány základní úlohy, v nichž si může student nastavit vstupní parametry a následně obdrží výsledek ve formě průběhů nebo číselných hodnot. Student tak upevňuje své znalosti problému díky praktické ukázce, kde může ovlivňovat vstupy a odhadovat výstupy.

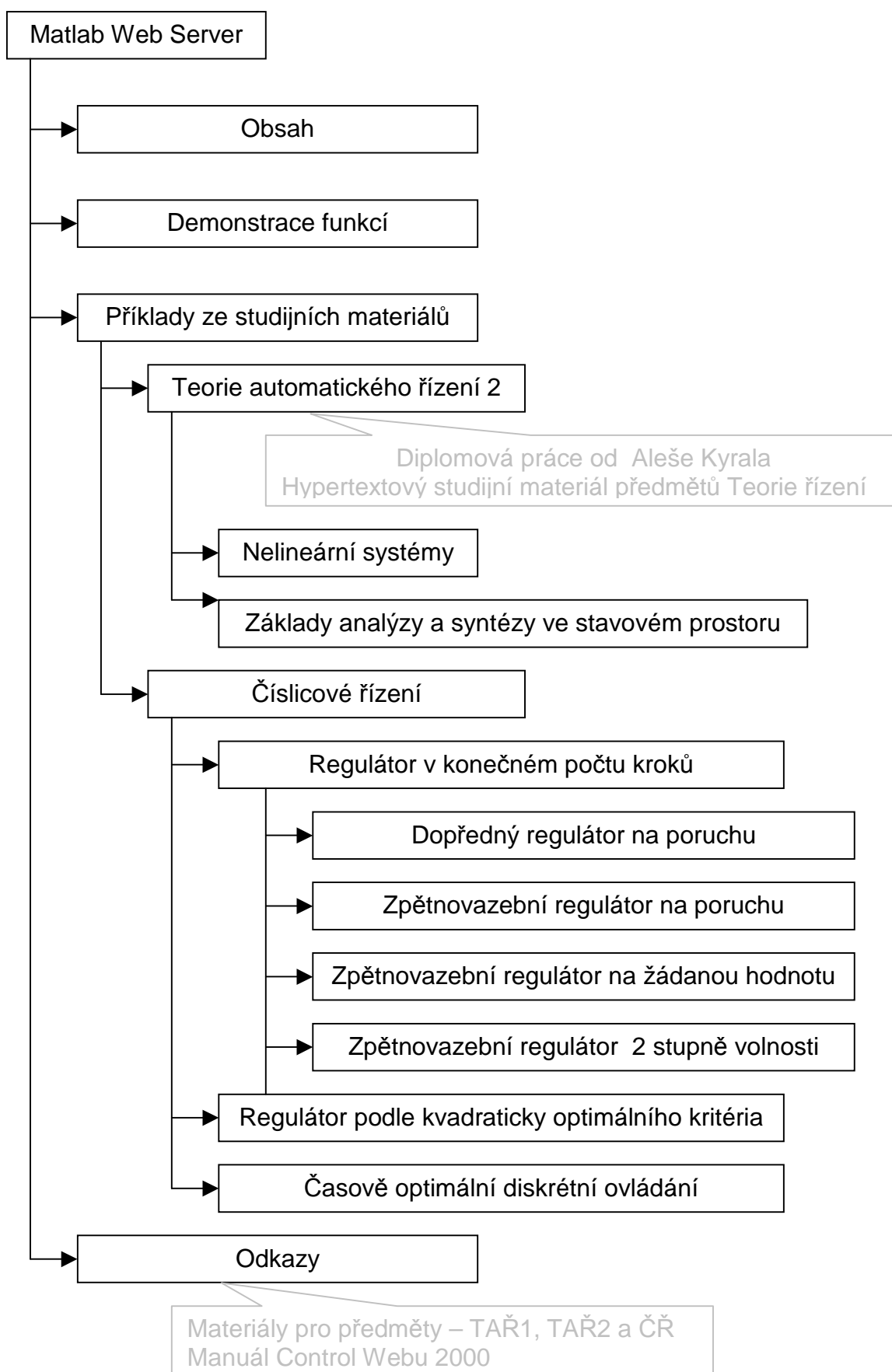
Struktura předmětu se může časem pozměnit, a tak bude třeba přidávat nová skripta, kvízy a interaktivní příklady. Tato diplomová práce by měla tento problém co nejvíce zjednodušit.

1. Struktura internetových stránek

Na následujícím obrázku je popsána struktura internetových stránek vytvořených pro podporu předmětu „Číslicové řízení“ a zjednodušeně i struktura stránek k nim vedoucích.



Obr. 1-1 Struktura internetových stránek předmětu „Číslicového řízení“



Obr. 1-2 Struktura internetových stránek, na kterých jsou realizovány příklady v Matlab web serveru

2. Předmět „Číslicové řízení“

Předmět „Číslicové řízení“ rozšiřuje vědomosti z předmětu „Teorie automatického řízení“, respektive jeho části, která se zabývá základy číslicového řízení.

Student se má seznámit s diskrétním popisem spojitých systémů, jak spojité systémy diskretizovat (Z-transformace váhové funkce atd.) a s odhadem parametrů diskrétního modelu systému z měření (struktura modelů ARX, ARMAX, BJ, OE), pomocí metody nejmenších čtverců.

V další části se provádí deterministická syntéza číslicového regulátoru pro systémy s jedním vstupem a jedním výstupem. Regulátory se navrhují pomocí dvou metod: 1) Regulace v konečném počtu kroků (silná a slabá verze)

2) Regulátor podle kvadraticky optimálního kritéria

Pomocí obou metod realizujeme regulátory pro vyrovnaní determinovaných poruch a změn žádané hodnoty (regulátory a jedním nebo dvěma stupni volnosti).

Na závěr se student seznámí se základy stavového řízení, estimací stavových veličin (Kalmanův estimátor) a návrhem stavového regulátoru (Linear-Quadratic-Gaussian (LQR) Control)

Ve cvičeních si student na reálné soustavě ověří přednášenou látku. Zadanou úlohu nejdříve identifikuje, aby získal diskrétní popis soustavy, a poté provede výpočty a odzkoušení jednotlivých regulátorů. Porovnáním různých metod výpočtu regulátoru získá student přehled o výhodách a nevýhodách jednotlivých postupů výpočtu.

Na tento předmět přímo navazují a teoretické znalosti studenta dále rozšiřují tyto předměty:

Řízení ve stavovém prostoru	9.sem.
Identifikace systémů	9.sem.
Metody optimálního řízení	10.sem.
Adaptivní a stochastické řízení	10.sem.

Internetové stránky předmětu „Číslicové řízení“:

http://www.fm.vslib.cz/~krtslib/fm/cir/krt_cir8.htm

3. Instalace

Na počítači musí být nainstalován operační systém na bázi NT (například Windows 2000 apd.) nebo Unix, pokud použijeme jiný operační systém, nepůjde nainstalovat Matlab Web Server (MWS). Pro zprovoznění serveru je třeba další program, v našem případě Apache HTTPD Server. Jelikož budeme na některých stránkách používat programovací jazyk PHP, musíme ho též nainstalovat na server.

3.1 Apache Httpd Server



Jedná se o nejrozšířenější program pro vytvoření a správu serverů. Jde o freeware program, tedy volně šiřitelný program. Instalační soubory je možno stáhnout na stránce autora <http://www.apache.org/> v sekci download. Server je možné provozovat pod operačním systémem Unix nebo Windows. Na počítači je nainstalován systém Windows 2000 a volíme tedy verzi pro Windows.

Pro instalaci programu není třeba žádných speciálních znalostí ohledně správy serverů. Program nás sám provede instalací. Na disku C:\ vytvoříme adresář **mws** a v něm podadresář **cgi-bin**. Před samotným spuštěním serveru však musíme provést některé změny v konfiguračním souboru **httpd.conf**, který se nejčastěji nalézá v adresáři:

C:\Program Files\Apache Group\Apache\conf.

ServerAdmin	e-mailová adresa správce serveru
ServerName	jméno počítače, na kterém je server umístěn
DocumentRoot	cesta k adresáři, ve kterém jsou soubory určené pro internetové stránky, stejnou cestu zapíšeme i do parametru <code>Directory</code> o několik řádků níže
DirectoryIndex	zde napíšeme soubory, které se otevřou při zadání adresy serveru, pokud neexistuje ani jeden ze souborů, dojde k chybovému hlášení (například <code>index.html index.htm</code> – první se pokusí otevřít <code>index.html</code> a poté <code>index.htm</code>)

ScriptAlias /cgi-bin/ zde napíšeme cestu k adresáři cgi-bin, který je podadresářem adresáře v parametru DocumentRoot

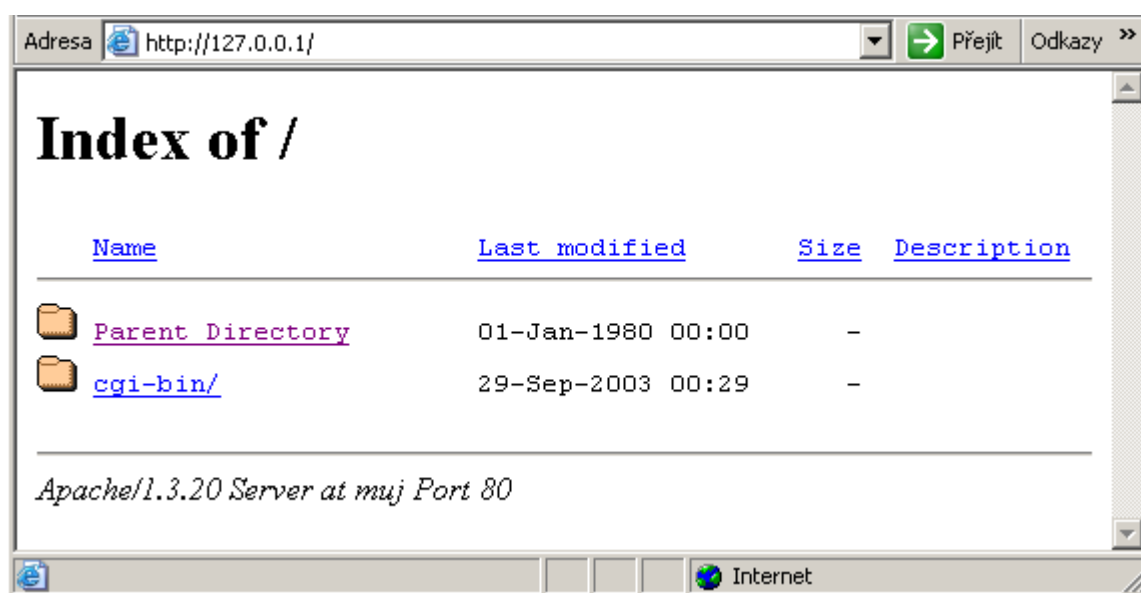
V našem případě budou jednotlivé parametry v souboru **httpd.conf** vypadat následovně.

```
řádek 237  ServerAdmin pavel.sir@centrum.cz
řádek 255  ServerName krt99
řádek 263  DocumentRoot "C:/MWS"
řádek 290  <Directory "C:/MWS">
řádek 349  DirectoryIndex index.html index.htm
řádek 560  ScriptAlias /cgi-bin/ "C:/MWS/cgi-bin/"
```

Pro zkušené správce serverů je v konfiguračním souboru mnoho parametrů pro nastavování. Podrobnějším popisem se však nebudu zabývat, jelikož je to pro naše účely nepotřebné.

Po uložení konfiguračního souboru je třeba server restartovat. Klikneme na **Start -> Nastavení -> Ovládací panely -> Nástroje pro správu -> Služby**. Nalezneme službu s názvem Apache a po kliknutí pravým tlačítkem myši se zobrazí menu, kde vybereme položku *Restartovat*.

Nyní by měl být server funkční. V internetovém prohlížeči zadáme adresu <http://127.0.0.1>. Jedná se o lokální adresu a pokud server pracuje správně, pak se v okně prohlížeče objeví obsah adresáře C:\MWS.



Obr. 3.1-1 Obsah internetového prohlížeče po zadání lokální adresy.

Adresa pro vnější uživatele (pokud jsme připojeni k síti) je odvozena od IP adresy daného počítače. V našem případě je tedy adresa <http://147.230.128.46/>. Server není zálohován, a pokud dojde k vypnutí počítače, nedojde k přesměrování na jiný server, ale stránky budou nefunkční.

3.2 Matlab Web Server



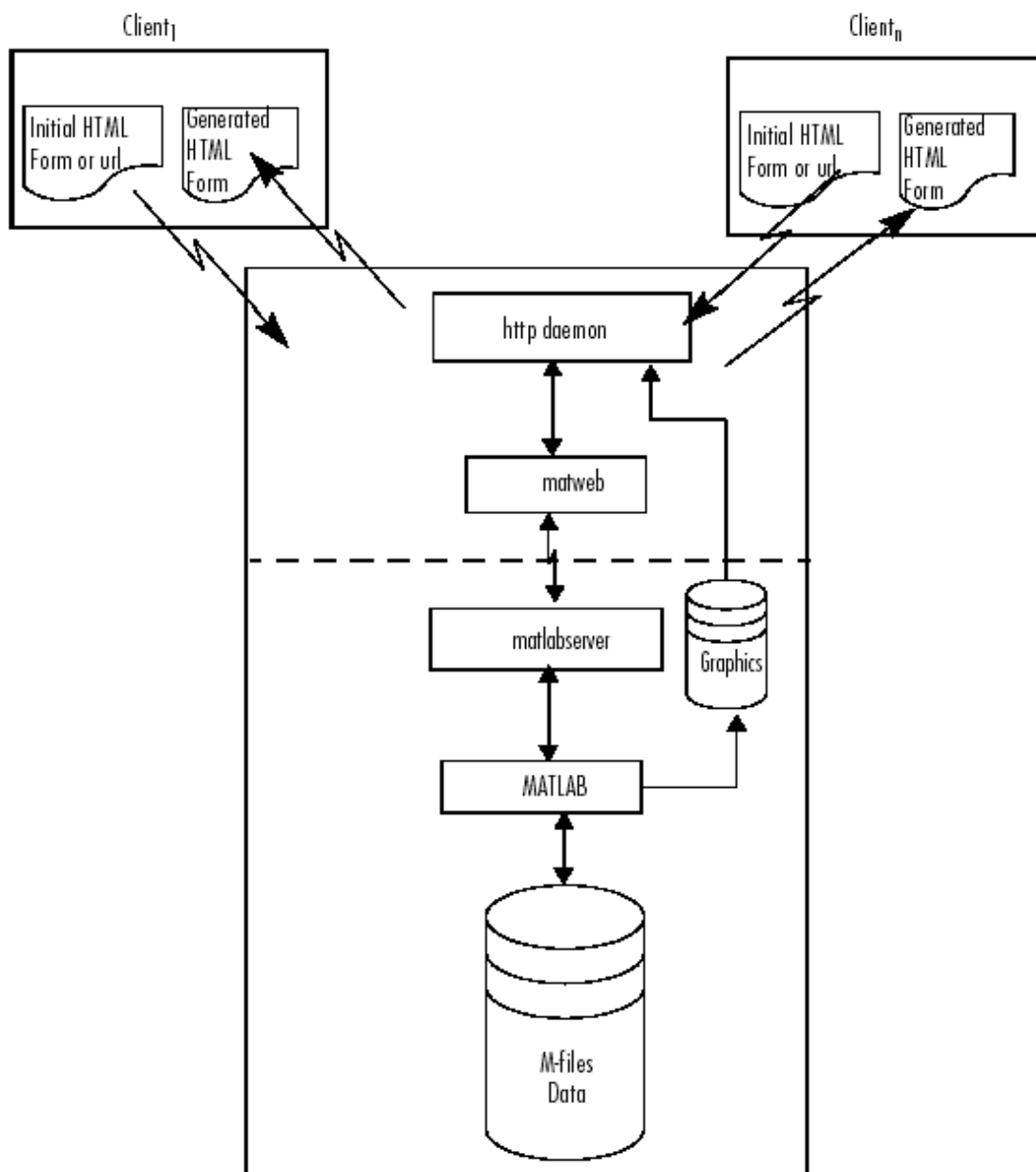
Matlab Web Server je standardně dodáván s programem Matlab jako jeho volitelná část při instalaci. Je třeba mít operační systém na bázi NT (v našem případě Windows 2000) nebo Unix a již zprovozněný server (viz. kapitola 3.1 Apache Httpd Server). Při instalaci Matlabu zkontrolujeme, zda je zaškrtnuta instalace Matlab Web Serveru. V případě, že nebyla dodržena některá z předchozích podmínek, nebude možno položku Matlab Web Server označit.

Při instalaci Apache Httpd Serveru jsme vytvořili adresář `C:\MWS\cgi-bin`, do tohoto adresáře nakopírujeme soubor **matweb.exe**, který je umístěn v adresáři s programem Matlab (například `C:\Matlab\webserver\bin\`). V adresáři `C:\MWS\cgi-bin` vytvoříme soubor **matweb.conf**. Do tohoto souboru budeme přidávat položky, vždy při vytvoření nové úlohy pro Matlab Web Server. (viz. kapitola 5. Vytvoření úloh pro Matlab WEB Server).

V adresáři s nainstalovaným Matlabem (například `C:\Matlab\webserver\`) se nalézá soubor **matlabserver.conf**. V souboru je jediný parametr `-m` „číslo“. Číslo označuje počet uživatelů, které může server najednou obsloužit. Je třeba volit mezi možnostmi obsluhy více uživatelů nebo výpočetní kapacitou. V našem případě jsem ponechal hodnotu 1, s ohledem na výpočetní kapacitu. Dalším parametrem může být `-p` „číslo portu“, který nastavuje číslo portu, kde Matweb „naslouchá“. Implicitně je zadán port 8888.

Podrobnější popis instalace a práce s Matlab Web Serverem je možné najít v dokumentaci **webserver.pdf**, která se nalézá v adresáři s nainstalovaným Matlabem `..\Matlab\help\pdf_doc\webserver\`. Ukázkové příklady je možno nalézt v adresáři `..\Matlab\toolbox\webserver\wsdemos`.

Následující schéma ukazuje, jak Matlab operuje nad Webem. Schéma je vytvořeno při možnosti přístupu dvou a více uživatelů (viz. parametr `-m` v souboru **matlabserver.conf**)



Obr. 3.2-1 Matlab nad Webem (převzato z manuálu k Matlab Web Serveru)

3.3 PHP



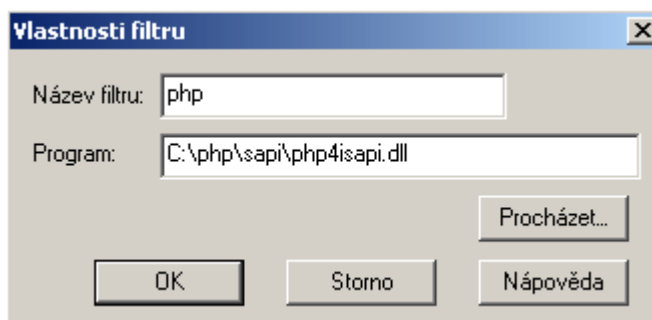
Skriptovací jazyk PHP budeme potřebovat kvůli internetovým stránkám s kvízy. PHP skript je zde použit pro zadávání a vyhodnocování odpovědí studentů. Jedná se o volně šiřitelný produkt (freeware), který je možno stáhnout ze stránky <http://www.php.net> ze sekce download. Opět jsou k dispozici verze pro Windows a Unix. Pro spuštění PHP je třeba mít nainstalovanou službu IIS (Internetová informační služba), která je součástí systému Windows. Pro instalaci této služby klikneme na **Start -> Nastavení -> Ovládací panely -> Přidat nebo ubrat programy**

-> **Přidat nebo odebrat součásti systému** a v okně *Součásti systému Windows* zaškrtneme **Internetová informační služba**. Potvrdíme stiskem tlačítka *Další*.

Na disku vytvoříme adresář PHP a do něj zkopírujeme instalaci jazyka PHP (například C:\PHP). V tomto adresáři nalezneme soubory `php.ini` a `php4ts.dll` a zkopírujeme je do adresáře s operačním systémem (například C:\Windows). V souboru `php.ini` provedeme následující změnu.

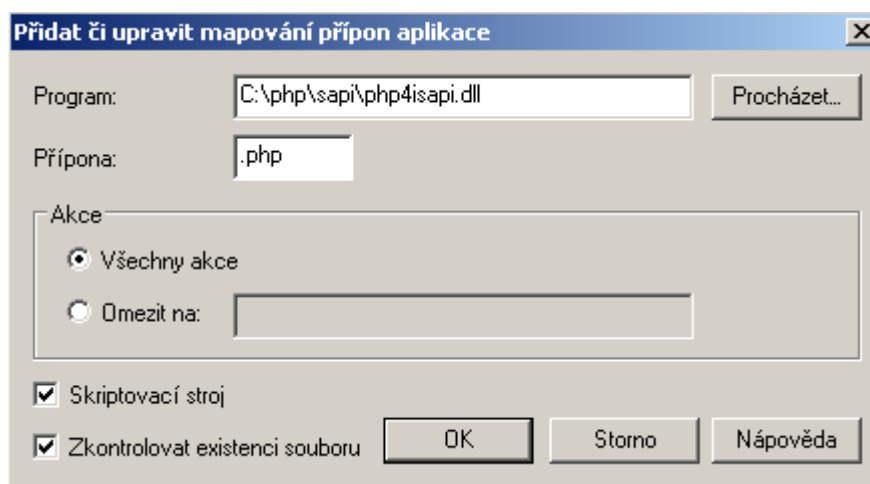
řádek 358 `extension_dir = C:\Php\extensions`

V další fázi musíme nakonfigurovat výchozí server WWW. Klikneme na **Start -> Nastavení -> Ovládací panely -> Nástroje pro správu -> Internetová informační služba**. Rozbalíme nabídku s názvem našeho počítače (krt99), klikneme pravým tlačítkem na **Výchozí webový server** a zvolíme *Vlastnosti*. Na záložce **Filtrování ISAPI** zvolíme *Přidat...* a položky vyplníme podle následujícího obrázku.



Obr.3.3-1 Nastavení filtrů ISAPI

Na záložce **Domovský adresář** zvolíme *Konfigurace...* a přidáme novou aplikaci.



Obr. 3.3-2 Konfigurace domovského adresáře

Na konec souboru `httpd.conf` (viz. kapitola 3.1 Apache Httpd Server) přidáme následující řádky.

```
ScriptAlias /php/ "c:/php/"  
AddType application/x-httpd-php .php .html .htm  
Action application/x-httpd-php "/php/php.exe"  
LoadModule php4_module c:/php/sapi/php4apache.dll
```

Pro zprovoznění PHP již stačí pouze službu Apache zrestartovat.

Podrobnější popis instalace (v angličtině) lze nálezt v souboru `install.txt` v adresáři `C:\PHP`.

4. Tvorba internetových stránek

O tvorbě WWW stránek již bylo napsáno mnoho textů (viz. Literatura [3],[4],[5]) a podrobnější popisování daného problému není potřeba, proto se zde budu zabývat pouze částí nezbytnou pro vytvoření jednoduchého formuláře na www stránce a problémy, které nastávají v různých internetových prohlížečích. Pro vytvoření webových stránek byl použit program Golden HTML editor. Program je volně k dispozici na adrese <http://golden.zde.cz/>, je však třeba odeslat autorovi žádost o registrační klíč, který je zaslán zdarma. Program je kompletně v češtině.

4.1 Deklarace formuláře

<FORM action="url_skriptu" method=[get | post]>

...prvky formuláře...

</FORM>

Párová značka vymezující formulář, kterým se získávají informace od uživatele. Formuláře nelze vnořovat.

action

URL skriptu, který zpracuje data odeslaná pomocí formuláře.

V našem případě bude data zpracovávat program `matweb.exe`.

method

Způsob odeslání dat.

get Doplněním na konec aktivního URL (implicitní). Pokud jsou odesílaná data delší než maximálně povolená délka URL dochází k problémům.

post V těle požadavku. Není třeba kontrolovat max. délku URL. Matlab Web Server pracuje s touto metodou.

4.2 Prvky formuláře

Prvky na formuláři mohou být různého druhu. Nejčastěji jsou používána textová pole, textová pole jako hesla, zaškrťovací políčka, různé druhy tlačítek a podobně. Kompletní přehled prvků formuláře lze najít v nápovědě k programu Golden HTML editor. V následující části textu jsou některé prvky formuláře ukázány na příkladu.

textové pole se jménem „Prijmeni“, implicitní hodnotou „Šír“ a velikostí 20

```
<input type="text" value="Šír" name="Prijmeni" size="20">
```

roletový výběr z možnosti „Muž“ nebo „Žena“, odesílá se „moznost1“ (resp. „moznost2“)

```
<select name="pohlavi">
  <option value="moznost1"> Muž
  <option value="moznost2"> Žena
</select>
```

výběrová políčka, ze všech políček se stejným názvem lze vybrat pouze jedno

Žák: `<input type="radio" name="druh" checked value="zak">`

Učitel: `<input type="radio" name="druh" value="ucitel">`

Žák: ☒ Učitel: ☐

zaškrťovací políčko, pokud je příkaz checked je implicitně zaškrtnuto

RC: `<input type="checkbox" name="RC">`

Fakultu: `<input type="checkbox" checked name="Fakultu">`

RC: ☐ Fakultu: ☒

tlačítko pro nastavení původních hodnot vstupů

```
<input type="reset" value="Původní hodnoty">
```

tlačítko pro odeslání formuláře

```
<input type="submit" value="Odeslat">
```

4.3 Nekompatabilita internetových prohlížečů

Při programování www stránky jsem několikrát narazil na problém v Html jazyce, se kterým si prohlížeč Internet Explorer dokázal poradit, ale například Netscape nikoliv. V některých případech je problém v typu operačního systému.

Jedním z problémů je správné napsání internetové adresy. Pro ukázkou použijeme soubor `materialy.htm` a internetové adresy:

- a) <http://www.fm.vslib.cz/~krtsub/fm/cir/materialy.htm>
- b) <http:\\www.fm.vslib.cz\\~krtsub\\fm\\cir\\materialy.htm>
- c) <http://www.fm.vslib.cz/~krtsub/fm/cir/MATERIALY.htm>

V případě a) je napsána adresa, kterou je možné otevřít ve všech odzkoušených internetových prohlížečích, jedná se o správně napsanou adresu. V případě, že otočíme lomítka v adrese (viz. b) nebo použijeme název souboru s rozdílnou velikostí písmen (viz. c), je tyto adresy možno otevřít pouze v prohlížeči Internet Explorer. Tento problém se neprojevuje pouze u odkazů na jiné stránky (htm,html apd.), ale i tehdy, pokud chceme použít na stránce obrázek nebo jiný dokument.

Problém, který je způsoben typem internetového prohlížeče nebo operačního systému, je používání různých fontů písma na stránce. V mém případě jde o používání fontu `Symbol`. Použijeme-li příkaz `m-n`, očekáváme na stránce text μ - ν , ale například v prohlížeči Netscape dojde k zobrazení textu **m-n**. Proto je lepší místo přetypování pomocí instrukce font napsat **μ-&nu**;

Řecká abeceda				Matematické operace			
příkaz	znak	příkaz	Znak	příkaz	znak	příkaz	znak
<code>&gamma;</code>	γ	<code>&Gamma;</code>	Γ	<code>&perp;</code>	\perp	<code>&ne;</code>	\neq
<code>&delta;</code>	δ	<code>&Delta;</code>	Δ	<code>&plusmn;</code>	\pm	<code>&exist;</code>	\exists
<code>&lambda;</code>	λ	<code>&Lambda;</code>	Λ	<code>&or;</code>	\vee	<code>&forall;</code>	\forall
<code>&xi;</code>	ξ	<code>&Xi;</code>	Ξ	<code>&and;</code>	\wedge	<code>&sube;</code>	\subseteq
<code>&phi;</code>	ϕ	<code>&Phi;</code>	Φ	<code>&le;</code>	\leq	<code>&isin;</code>	\in
<code>&omega;</code>	ω	<code>&Omega;</code>	Ω	<code>&ge;</code>	\geq	<code>&harr;</code>	\leftrightarrow

Tab. 4.3-1 Vybrané symboly a příkazy používané na internetové stránce

Kompletní seznam používaných symbolů (v angličtině) je možno najít na adrese <http://www.w3.org/Arena/tour/symbols.html>.

5. Vytvoření úloh pro Matlab Web Server

Vytvoření úlohy pro Matlab Web Server se dá rozdělit na několik úkolů:

1) Analýza úlohy

- Stanovení cílů úlohy
Co přesně chceme získat (např. graf, hodnoty veličin v určitém čase atd.)
- Definování vstupních a výstupních dat
Jaká data bude zadávat dotazovatel a jaká data je potřeba mít na výstupu.

2) Vstupní stránka

Podle analýzy úlohy vytvoříme HTML dokument, který nám umožní zadávání vstupních dat.

3) Výstupní stránka

Jedná se opět o HTML dokument, pomocí kterého zobrazujeme výsledná data (hodnoty veličin, obrázky atd.).

4) M-file

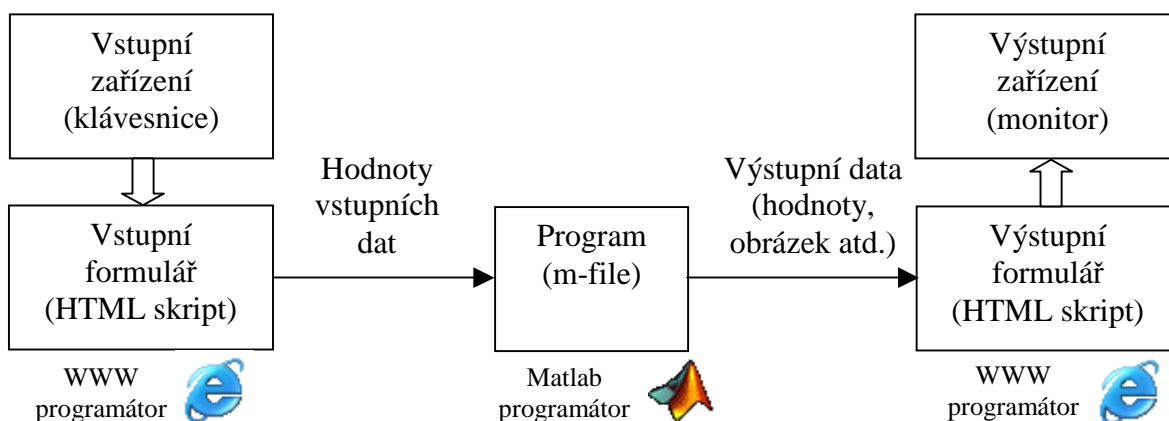
Program na výpočet, realizovaný pomocí Matlab Web Serveru.

Načítá data z vstupního formuláře, zpracovává a odesílá data na výstupní formulář.

vstupní formulář:	test_in.html
m-file:	test.m
výstupní formulář:	test_out.html

Tab.5-1 Pojmenování souborů v C:\Mws\fce

Při řešení složitějších úloh je možné, aby každou část programoval někdo jiný, ale musí být kladen důraz na analýzu úlohy.



5.1 Vstupní stránka

Jde o formulář, pomocí kterého zadáváme data nutná pro výpočet, prováděný pomocí m-filu.

V našem případě nejvíce používáme textová pole pro zadávání polynomů (ve formě vektorů), popřípadě jiné číselné údaje (doba simulace atd.). Jelikož u všech úloh očekáváme obrázek průběhů veličin, pomocí zaškrtačích políček si navolíme, které průběhy se mají zobrazit. Nejdůležitější průběhy jsou předvoleny.

Vstupní data bude zpracovávat m-file, který lze spustit pomocí programu Matlab Web Server. Párová značka form bude mít tedy následující hodnoty.

```
<form action=" ../cgi-bin/matweb.exe" method="post">
```

↳ umístění souboru viz. kapitola 3-2 Matlab Web Server

Formulář musíme doplnit o skrytý vstup, který určuje m-file pro zpracování dat.

```
<input type="hidden" name="mlmfile" value="ukazkova_uloha">
```

↳ typ souboru ↳ název souboru

Přenosy:

A: B: H:

Žádaná hodnota:

čas vzniku:

Porucha:

velikost skoku: čas vzniku:


Obecné:

vzorkovací čas: doba simulace:

Které grafy chcete zobrazit:

u: ☐ y: ☒ d: ☒ w: ☒

[Zpět](#) | [Úvod](#)

 [Fakulta Mechatroniky a mezioborových inženýrských studií](#)
[Technická univerzita v Liberci - Katedra řídicí techniky](#)
 Autor: [Pavel Šír](#)

Obr. 5.1-1 Ukázka vstupního stránky pro zadání dat

HTML skript vstupní stránky vypadá takto:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<html>
<head>
  <META HTTP-EQUIV="Content-Type" CONTENT="text/html; CHARSET=Windows-1250">
  <TITLE>Matlab WEB Server - TU Liberec - CR</TITLE>
</head>
<body bgcolor="white">
<form action=" ../cgi-bin/matweb.exe" method="post">
  <center>
    <input type="hidden" name="mlmfile" value="ukazkova_uloha">
    <u>Přenosy:</u>
    <table>
      <tr><td>A: <input type="text" name="A" value="[1 -1 1]" size="20">
      <td>B: <input type="text" name="B" value="[0 1 0.5]" size="20">
      <td>H: <input type="text" name="H" value="[1 -1]" size="20">
    </table><br>
    <u>Žádaná hodnota:</u><br>
    <table>
      <tr><td>čas vzniku: <input type="text" name="wstart" value="1" size="10">
    </table><br>
    <u>Porucha:</u><br>
    <table>
      <tr><td>velikost skoku: <input type="text" name="damp" value="1" size="10">
      <td>čas vzniku: <input type="text" name="dstart" value="15" size="10">
    </table><br>
    <u>Obecné:</u><br>
    <table>
      <tr><td>vzorkovací čas: <input type="text" name="sample" value="1" size="10">
      <td>dobu simulace: <input type="text" name="doba" value="30" size="10">
    </table> <br>
    <u>Ktere grafy chcete zobrazit:</u>
    <table>
      <tr><td>u:<input type="checkbox" name="u">
      <td>y:<input type="checkbox" checked name="y">
      <td>d:<input type="checkbox" checked name="d">
      <td>w:<input type="checkbox" checked name="w">
    </table><br><br>
    <input type="submit" value="odeslat">
  </form><br>
  <a href=" ../uvod.html">Zpět</a>&nbsp;&nbsp;&nbsp;<a href=" ../index.html">Úvod</a>
<p align=right>
  <a href=http://www.vslib.cz>
  <img src=../img/TU.gif align=left height=50 border=0></a>
  <a href=http://www.fm.vslib.cz/~krt/czframe.htm>
  <img src=../img/KRT.gif align=left height=50 border=0></a>
  <font size=-1>
  <a href=http://www.fm.vslib.cz>
    Fakulta Mechatroniky a mezioborových inženýrských studií</a><br>
  <a href=http://www.vslib.cz>Technická univerzita v Liberci</a> -
  <a href=http://www.fm.vslib.cz/~krt/czframe.htm>Katedra řídicí techniky</a><br>
  Autor: <a href=mailto:pavel.sir@centrum.cz>Pavel Šír</a></font></p>
</body>
</html>
```

Stránku lze doplnit o další upřesňující informace, jako je zadání úlohy, schéma zapojení, základní vztahy a další údaje pro větší názornost.

5.2 Výstupní stránka

Na této stránce se budou zobrazovat data vstupující do výpočtu (m-filu), data vypočtená a obrázky průběhů.

Vstupní a výstupní stránka jsou téměř totožné, ale na výstupní stránce musíme zobrazovat data použitá ve výpočetním m-filu. Hodnoty dat zobrazíme pomocí názvu proměnné použité v m-filu, kterou vepíšeme mezi znaky \$.

Ukázka zobrazení dat v textovém poli:

Vstupní formulář

```
<td>B: <input type="text" name="B" value="[0 1 0.5]" size="20">
```

↳ implicitní hodnota

Výstupní formulář

```
<td>B: <input type="text" name="B" value="$B$" size="20">
```

↳ zobrazí hodnotu proměnné B

Obdobně lze zobrazovat hodnoty i mimo prvky formuláře. Například pokud chceme zobrazit výsledný obrazový přenos.

```
<table><tr><td>F(z) = <td align=center>[$B$]<hr>[$A$]</table>
```

Musíme dávat pozor na správné názvy proměnných použitých v m-filu. Chceme-li zobrazovat proměnné použité v m-filu, pak se místo \$B\$ zobrazí hodnota proměnné B používaná v m-filu, v opačném případě nikoliv.

Proměnné B a A jsou známe Proměnné B a A jsou neznámé

$$F(z) = \frac{\begin{bmatrix} 0 & 1 \end{bmatrix}}{\begin{bmatrix} 1 & 2 \end{bmatrix}}$$

$$F(z) = \frac{\begin{bmatrix} \$B\$ \end{bmatrix}}{\begin{bmatrix} \$A\$ \end{bmatrix}}$$

Podobně jako hodnoty proměnných lze zobrazovat i obrázky (průběhy).

```

```

↳ umístění a název souboru s obrázkem
(název souboru je jako proměnná)

Pokud budeme spouštět program na počítači, kde je nainstalován Matlab Web Server (testování a vývoj), můžeme použít lokální IP adresu 127.0.0.1, v provozu však musíme použít IP adresu serveru. Umístění zadáváme pouze ve formátu

127.0.0.1/fce, jelikož adresář C:\MWS je nastaven jako pracovní adresář pro Matlab Web Server (viz. kapitola 3-2 Matlab Web Server).

5.3 Zpracování na straně serveru (m-file)

Na vstupní stránce se pomocí vstupu s názvem mlmfile a parametrem, kterým je název volaného souboru, odkazujeme na m-file pro zpracování vstupních dat (viz. kapitola 5-1 Vstupní stránka).

5.3.1 Načítání a odesílání dat

Pro lepší pochopení důležitých instrukcí v m-filu si vytvoříme příklad pro zobrazení přechodové charakteristiky systému, který získáme z polynomů A a B, zadávaných z textového pole jako dva vektory hodnot. Soubor bude mít název test.m

```
function HTMLout=test(h)    %vytvoření funkce s názvem souboru
mlid=getfield(h,'mlid');    %načtení vstupních dat do proměnné h
cd(h.mldir);               %nastavení pracovního adresáře
                           %jeho umístění je definováno v souboru
                           %matweb.conf (viz. kapitola 3-2 Matlab
                           %Web Server)

B=h.B;                     %uložení proměnné B ze vstupního řetězce h do B
                           %název proměnné definované na vstupní stránce je B
A=h.A;

s.B=B;                     %uložení proměnných do výstupního řetězce
                           %pouze chceme-li je zobrazovat na výstupní stránce
s.A=A;

sys=tf(eval(B),eval(A));    %vytvoření systému
                           %eval-převádí řetězec na číslo (vektor)
step(sys);                 %vykreslení přechodové charakteristiky



savejpg1=strcat('img',num2str(round(cputime^cputime*1000000)));
                           %generování náhodného jména souboru
                           %strcat- spojuje řetězce do jednoho
savejpg1=strcat(savejpg1,'.jpg');
                           %soubor bude mít příponu jpg
s.img1=savejpg1;           %uložení názvu obrázku do řetězce s
wsprintfjpeg(gcf,savejpg1); %vytvoření obrázku ve formátu jpg
outfiletemp=which('test_out.html'); %název výstupního formuláře
                           %kde zobrazíme data

HTMLout=htmlrep(s,outfiletemp);
                           %ve výstupním formuláři se místo $proměnná$ zobrazí hodnoty
                           %proměnných uložených v řetězci s
```

Definování názvu obrázku nelze provést pomocí funkce `rand`, protože tato funkce vrací stejné hodnoty při opětovném spuštění Matlabu. Jelikož používáme program Matlab Web Server, který se pro každé volání spouští znovu, měly by všechny soubory stejný název a obrázky by se neustále přepisovaly. V případě, že bychom chtěli obrázky archivovat, by to nebylo možné. Proto používáme hodnotu `cputime`, při které očekáváme různé doby zpracovávání m-filu a tedy i různé názvy obrázků.

V následující tabulce jsou popsány instrukce pro získání hodnot z různých prvků formuláře. Hodnoty načítané v m-filu jsou uloženy v řetězci „h“.

`mlid=getfield(h, 'mlid')` – načtení vstupních dat do proměnné h

Prvky formuláře	Získání hodnoty v m-filu
<code><input type="text" value="[1 -0.5]" name="A"></code> 	<code>A=h.A</code>
<code><select name="pohlavi"></code> <code> <option value="moznost1"> Muž</code> <code> <option value="moznost2"> Žena</code> <code></select></code> 	<pre>switch h.pohlavi case 'moznost1' str='muz'; case 'moznost2' str='zena'; end</pre>
<p>Žák:</p> <code><input type="radio" name="druh" checked value="zak"></code> <p>Učitel:</p> <code><input type="radio" name="druh" value="ucitel"></code> <p>Žák: <input checked="" type="radio"/> Učitel: <input type="radio"/></p>	<pre>switch h.druh case 'zak' prace='zak'; case 'ucitel' prace='ucitel'; end</pre>
<p>y: <code><input type="checkbox" checked name="y"></code> y: <input checked="" type="checkbox"/></p>	<pre>if isfield(h, 'y') vybrano=1; end;</pre>

Tab. 5.3.1-1 Jak získat data z různých prvků formuláře

5.3.2 Simulace úloh

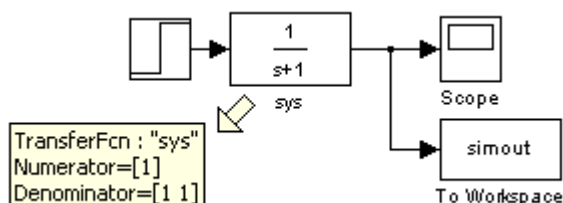
V úlohách pro předmět „Číslicové řízení“ budeme chtít úlohy i simulovat pomocí simulinkového modelu. Schémata vytvořená v různých verzích programu Matlab není možné zaměňovat, proto musí být schémata vytvořena ve verzi Matlabu, který je na serveru instalován. V našem případě se jedná o verzi 5.3.0.10183 (R11).

Simulaci si předvedeme na úloze pro zobrazení přechodové charakteristiky pomocí simulinku. Použijeme stejný program jako v příkladě vytvoření přechodové charakteristiky pomocí funkce `step` (viz. kapitola 5.3.1 Načítání a odesílání dat), pouze ho rozšíříme o několik instrukcí a vynecháme instrukci `step`. Soubor se simulinkovým modelem má název `sch_test.mdl` (Obr 5.3.2-1).

```
function HTMLout=test(h)
mlid=getfield(h,'mlid');
cd(h.mldir);
B=h.B;A=h.A;
s.B=B;s.A=A;

load_system('sch_test'); %načtení souboru *.mdl
set_param('sch_test/sys','Numerator',B); %čitatel
set_param('sch_test/sys','Denominator',A); %jmenovatel
sim('sch_test'); %spuštění simulace
%ze simulace vystupuje proměnná s názvem simout, která je
%uložena ve Workspace -> můžeme ji dále zpracovávat

plot(simout) %vykreslení průběhů
savejpgl=strcat('img',num2str(round(cputime^cputime*1000000)));
savejpgl=strcat(savejpgl,'.jpg');
s.imgl=savejpgl;
wsprintjpeg(gcf,savejpgl);
outfiletemp=which('test_out.html'); %název výstupní stránky
HTMLout=htmlrep(s,outfiletemp);
%odešle proměnné na výstupní stránku jako textový řetězec „s“
```



Obr. 5.3.2-1 Simulinkové schéma `sch_test.mdl`






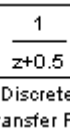


Hodnoty výsledného průběhu budou uloženy ve workspacu v proměnné `simout`. Použitím instrukce `plot` získáme průběh, který můžeme uložit již zmíněným postupem jako obrázek.

Příkaz `sim` lze rozšířit o některé atributy:

```
opt=simset('maxstep',eval(sample));
sim('sch_dopredny_1st_porucha',doba,opt)
```

Požadujeme-li nastavit maximální krok simulace, slouží k tomu proměnná `sample` a pro nastavení doby simulace proměnná `doba`. Obě tyto proměnné se zadávají na vstupní stránce pomocí textového pole.

Názvy parametrů, které chceme v simulinkovém modelu změnit, zjistíme při přejetí myší nad zvoleným objektem v prostředí simulink. Dáváme pozor na správné formáty dat, zapisovaných do parametrů modelu. Většinou se zapisují parametry jako čísla, a tak musíme textový řetězec pomocí příkazu `eval` převést na číslo.

Značka	parametry	popis parametrů
 Step	Time Before After Sample	Čas vzniku Počáteční hodnota Koncová hodnota Vzorkovací čas
 Constant	Value	Hodnota konstanty
 Sum	IconShape Inputs SaturateOnIntegerOverflow	Tvar bloku: rectangular nebo round Počet a druh vstupů: + nebo - Saturace při přetečení: on nebo off
 Gain	Gain SaturateOnIntegerOverflow	Hodnota zesílení Saturace při přetečení: on nebo off
 Unit Delay	X0 SampleTime	Počáteční podmínka Vzorkovací čas
 Discrete Transfer Fcn	Numerator Denominator SampleTime	Číselník Jmenovatel Vzorkovací čas
 Discrete Filter	Numerator Denominator SampleTime	Číselník Jmenovatel Vzorkovací čas
 To Workspace	VariableName Buffer Decimation SampleTime SaveFormat	Proměnná pro uložení hodnot Kolik hodnot zachytit: inf nebo číslo Vyber každou x-tou hodnotu Vzorkovací čas Typ výstupu: Matrix nebo Structure

Tab. 5.3.2-2 Nejčastěji používané bloky v prostředí Matlab-simulink

V modelu nesmíme zapomenout nastavit ve všech blocích stejný vzorkovací čas. Pro blok `simout` a `step` je však vhodné použít vzorkovací čas 0, jelikož v opačném případě může docházet k menším deformacím průběhů.

5.3.3 Zobrazení průběhů

Na vstupní stránce je možné pomocí zaškrtačacích políček navolit požadované průběhy. V m-filu je tedy nutné ošetřit všechny možnosti volby. Jako příklad si uvedeme úlohu, kde je možné zvolit průběhy veličin u , y , d .

```
x=0;
if (isfield (h, 'u')) Y(x+1,:)=Y(:,1)';
    leg(x+1,:)= 'u';
    %proměnná s názvy průběhů zobrazovaných v legendě
    Yd(x+1,:)= 'bo'; %proměnná do které ukládáme druhy čar průběhů
    x=x+1;
end;
if (isfield (h, 'y')) Y(x+1,:)=Y(:,2)';
    leg(x+1,:)= 'y';
    Yd(x+1,:)= 'ro';
    x=x+1;
end;
if (isfield (h, 'd')) Y(x+1,:)=Y(:,3)';
    leg(x+1,:)= 'd';
    Yd(x+1,:)= 'm ';
    x=x+1;
end;
switch x
case 1
    plot(tout,Y(1,:),Yd(1,:))
    legend(leg,4)
case 2
    plot(tout,Y(1,:),Yd(1,:),tout,Y(2,:),Yd(2,:))
    legend(leg,4)
case 3
    plot(tout,Y(1,:),Yd(1,:),tout,Y(2,:),Yd(2,:),tout,Y(3,:),Yd(3,:))
    legend(leg,4)
otherwise
    plot(0,0)
end
```

Pokud je například požadován průběh u a y , pak do proměnné Y (neplést s proměnnou y) do prvního řádku vložíme hodnoty z workspace odpovídající nasimulované hodnotě u . Musíme zadat název, barvu a druh čáry průběhu. To provádíme zapsáním požadovaných hodnot do proměnných leg a Yd opět do prvního řádku. Podobně postupujeme i pro přidávání průběhu y , ale veškeré hodnoty ukládáme na druhý řádek v proměnných. V proměnné x je počet průběhů, které budeme zobrazovat. Pomocí instrukce `switch`, pro různé hodnoty x vykreslíme průběhy (v našem případě $x=2$).

V případě, že bychom nezadávali barvu a styl čáry, mohli bychom příkaz `switch` vynechat a psát pouze `plot(tout,Y)` i pro více rozměrové matice Y .

Typy barev a stylů čáry získáme po zadání příkazu `help plot` v Matlabu.

5.3.4 Číslo nebo řetězec znaků

Je třeba kontrolovat typy proměnných. Nelze zpracovávat textový řetězec jako číslo a naopak nelze odeslat číslo jako textový řetězec.

Ze vstupní stránky načítáme textové řetězce. Pomocí instrukce `eval` je převedeme na číslo, s kterým lze dále v programu počítat. Pokud číslo odesíláme na výstupní stránku, použijeme instrukci `num2str`, která převádí číslo na textový řetězec.

```
A=eval(A)           %převede textový řetězec na číslo
A=num2str(A,2);      %převede číslo A na řetězec o dvou znacích
                    %desetinná čárka se nepočítá jako znak
```

Při simulování úlohy se zadávají parametry jednotlivých bloků jako textové řetězce a je velmi důležité zadat správný formát dat.

```
set_param('sch_u/sys','Numerator',strcat('[' ,num2str(B),']'))
    %proměnná B je číslo a je tedy nutné převést na textový řetězec
    %příkaz strcat spojuje několik textových řetězců do jednoho

set_param('sch_u/sys','Numerator',strcat('[' ,B,']'))
    %proměnná B je textový řetězec
```

5.3.5 Problém

Nelze používat instrukci `disp`, pokud realizujeme programy (m-fily) pomocí Matlab Web Serveru. Jestliže instrukci použijeme, dochází k chybě „500 Internal Server Error.“

Tento problém se projeví zejména, pokud chceme inicializovat polynomiální toolbox pomocí instrukce `pinit`, kterou vytvořila firma Polyx (www.polyx.cz). Kvůli tomuto problému jsem se elektronickou poštou spojil s firmou Polyx, ale v odpovědi firma napsala, že nezkoušela polynomiální toolbox v prostředí Matlab Web Server a může tedy docházet u instrukcí k chybovému hlášení.

Zajímavé je, že pokud zapomeneme v programu (m-filu) středník za některou instrukcí, výsledek se v internetovém prohlížeči objeví a nedojde k chybovému hlášení, jako v případě použití instrukce `disp`.

6. Toolbox CR

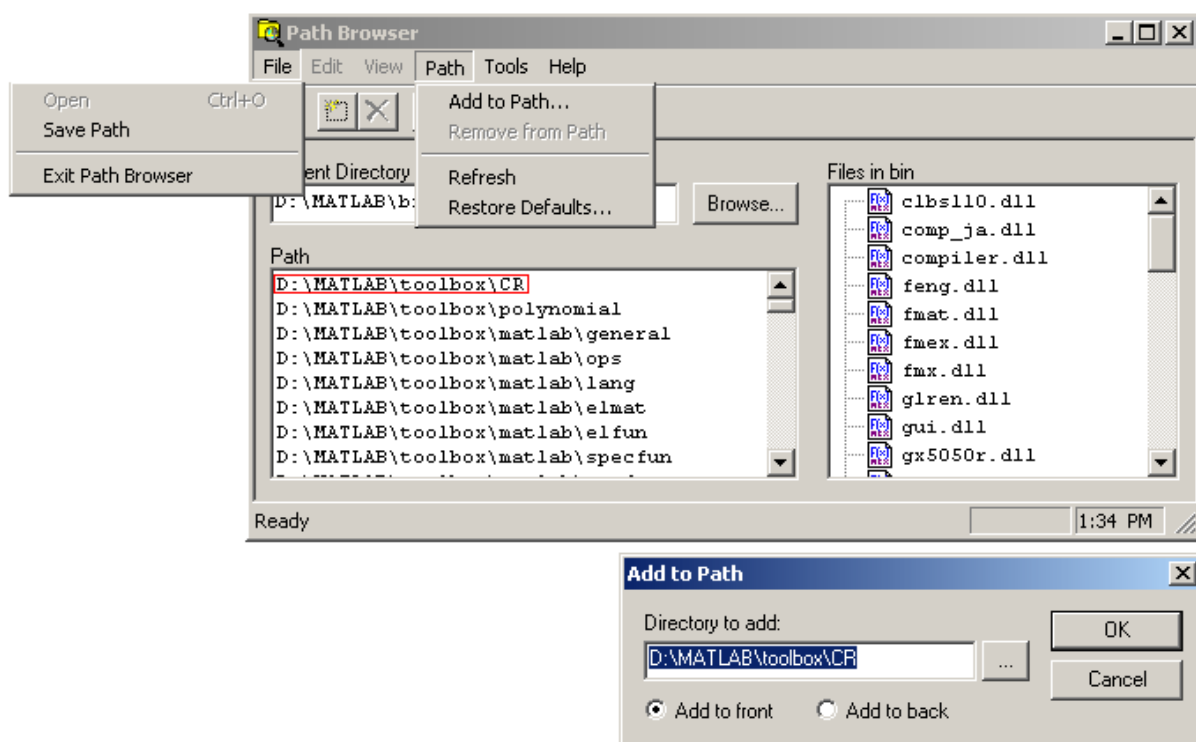
Tento toolbox byl vytvořen pro zjednodušení práce při programování v prostředí Matlab. Realizuje nejčastěji používané rutiny při výpočtech v předmětu „Číslicové řízení“. K jeho funkci je třeba mít nainstalovaný polynomial toolbox od firmy Polyx.

6.1 Instalace toolboxu CR

Většina toolboxů je instalována v adresáři `..\Matlab\toolbox`. Je tedy vhodné volit stejné umístění, i když je možné mít toolbox v libovolném adresáři.

V námi zvoleném adresáři vytvoříme adresář „CR“ a do něj nakopírujeme vytvořené m-fily. Poté zapneme Matlab a v základním okně spustíme **Path Browser -> Path -> Add to Path....** Do okénka **Directory to add** napíšeme zvolenou cestu k toolboxu (například `D:\Matlab\toolbox\CR`). Nebo pomocí tlačítka „...“ můžeme složku vyhledat. Po potvrzení by se v okně **Path** měla objevit námi zvolená cesta. Cesta se neobjeví, pokud adresář neexistuje. Na záložce **File** nesmíme zapomenout cesty uložit pomocí **Save Path**.

Po novém otevření Matlabu je možné volat funkce z toolboxu CR.



Obr. 6.1-1 Path Browser v programu Matlab

6.2 Funkce toolboxu CR

V následujícím textu popíšeme funkce v toolboxu CR a nejdůležitější problémy spojené s jejich programováním. Pro názornost uvedu ke každé funkci příklad, kde bude vidět, jaké parametry vstupují do funkce a jaké vystupují.

6.2.1 pinitserver.m

Obdobně jako originální soubor `pinit.m` (od firmy Polyx) inicializuje polynomiální toolbox. Jsou zde však vynechány příkazy:

```
warning on;
disp(' ');
disp(' Polynomial Toolbox initialized. To get started, type one of');
disp(' these: helpwin or poldesk. For product information, visit');
disp(' www.polyx.com or www.polyx.cz.');
```

Tyto příkazy způsobovaly chybové hlášení v internetových prohlížečích, pokud jsme inicializovali polynomiální toolbox v prostředí Matlab web serveru.

Internal Server Error

The server encountered an internal error or misconfiguration and was unable to complete your request.

Please contact the server administrator, pavel.sir@centrum.cz and inform them of the time the error occurred, and anything you might have done that may have caused the error.

More information about this error may be available in the server error log.

Tab. 6.2.1-1 Chybové hlášení při inicializování polynomiálního toolboxu pomocí instrukce `pinit`

6.2.2 pm.m

Ze zvoleného polynomu zjistí stabilní část, nestabilní část a dopravní zpoždění (možné vytknutí mocniny q). Rozdělení provádíme pomocí kořenů polynomiální funkce. Pokud je kořen:

0 – lze vytknout q (dopravní zpoždění)

≥ 1 – stabilní kořen

< 1 – nestabilní kořen

Napsaná kritéria platí pouze pro polynomy v mocninách $q = z^{-1}$.

Příklad:

$$A(q) = q(1 - 0,5q)(1 - 2q) = q - 2,5q^2 + q^3$$

```
» A=[0 1 -2.5 1]
» [Aplus, Aminus ,ny] = pm (A)
```

} zadávání v Matlabu

$$\left. \begin{array}{l} A_{\text{plus}} = 1 - 0.5q \\ A_{\text{minus}} = 1 - 2q \\ n_y = 1 \end{array} \right\} \text{výsledky v Matlabu}$$

Funkce využívá instrukci `roots` pro zjištění kořenů polynomu a poté kořeny porovnáváme podle zvoleného kritéria. Pokud kořen odpovídá danému kritériu například kořen ≥ 1 – stabilní kořen, je potřeba z kořenů opět složit polynom. To provádíme pomocí instrukce:

$$A_{\text{plus}} = A_{\text{plus}} * \text{mat2pol} (\text{fliplr} ([1 \ -1/\text{koreny}(i)]))$$

$\hookrightarrow q^0 \hookrightarrow q^1$

Polynom skládáme za předpokladu, že jsme vytknuli před polynom takové číslo, aby se $q^0=1$. Tímto číslem poté vynásobíme výsledek.

6.2.3 kraceni.m

Pokud je to možné, vykrátí dva polynomy mezi sebou.

Příklad:

$$\begin{aligned} A(q) &= q(1 - 0.5q)(1 - 2q) = q - 2.5q^2 + q^3 \\ B(q) &= q(3 - q)(1 - 2q) = 3q - 7q^2 + 2q^3 \\ \gg A &= q - 2.5q^2 + q^3 \\ \gg B &= 3q - 7q^2 + 2q^3 \\ \gg [A_k, B_k] &= \text{kraceni} (A, B) \\ A_k &= [1.0000 \ -0.5000] \quad \rightarrow 1 - 0.5q \\ B_k &= [3 \ -1] \quad \rightarrow 3 - q \end{aligned}$$

Opět využíváme kořenů polynomu, které mezi sebou krátíme.

V programu využíváme tři proměnné:

$$\begin{array}{ll} \text{nula} = -8; & \%s \text{ jakou presnosti urcujeme nulu} \\ \text{vysledek} = -5; & \%presnost \text{ zaokrouhleni vysledku} \\ \text{koreny} = -5; & \%presnost \text{ pri porovnani korenu} \end{array}$$

Proměnná `nula` má vliv na přesnost, podle které označíme číslo za nulové. Například $1.5000e-08$ je již z praktického hlediska 0 a nemá vliv na výsledek, proto je třeba takové výsledky správně interpretovat.

Proměnná `vysledek` pouze značí, s jakou přesností nám postačuje vypočtený výsledek, a na samotný proces krácení nemá žádný vliv.

Proměnná `koreny` udává, s jakou přesností porovnáváme kořeny dvou polynomů. Zde dochází k nejčastějším problémům, jelikož vlivem zaokrouhlování může docházet k chybám, kdy očekáváme některé stejné kořeny u dvou polynomů, ale při porovnání pomocí instrukcí `if` nezjistíme žádné shodné kořeny.

Přesnost na pět desetinných míst postačuje pro většinu aplikací, ale počítač kontroluje shodnost čísel až na délku svého formátu, kde však může dojít k chybě vlivem zaokrouhlení. Pro ukázkou:

číslo A = 1,123456789 a číslo B = 1,123456790

- pro naše potřeby můžeme prohlásit čísla za shodná
- při použití instrukce `if A==B` je počítač označí za různá

Nevýhodou tohoto řešení je různorodost výsledků. V různých aplikacích se mohou kořeny lišit na různých desetinných místech. Proto není správné zadávat proměnnou `koreny` jako konstantu. V případě výpočtů na webových stránkách předmětu „Číslicové řízení“ jsem nastavil proměnnou `koreny` tak, aby vyhovovala většině aplikací. Lepších výsledků by bylo možné dosáhnout metodou dynamického porovnávání, kdy bychom postupně snižovali přesnost, dokud bychom neobjevili dva shodné výsledky.

Tento problém je nejvíce vidět u výpočtů regulátorů, kdy očekáváme, že dojde k vykrácení polynomů (soudělné polynomy), ale vlivem nepřesnosti nelze polynomy zkrátit a výsledný regulátor je nefunkční.

Pro ukázkou možných chyb zvolím následující příklad:

(Teoretickou část příkladu zde nebudu popisovat, uvedu pouze získané výsledky.)

Regulace otáček tachodynamu spojeného se stejnosměrným motorem pružnou spojkou



Obr. 6.2.3-1 Blokové schéma úlohy

Úloha je realizována v učebně TK4 pavilonu A TU v Liberci.

Soustavu budeme regulovat pomocí zpětnovazebního regulátoru na žádanou hodnotu, vypočteného podle kvadraticky optimálního kritéria.

V prostředí Matlab jsme provedli měření odezvy na několik skoků žádané hodnoty v okolí pracovního bodu. Z tohoto měření jsme mohli soustavu identifikovat.

$$F(s) = \frac{2.109}{0.07952s^3 + 0.4525s^2 + 1.033s + 3.501}$$

Při vzorkovací čase $T=0,05s$ získáme diskrétní přenos

$$F(s) = \frac{0.0005146z^2 + 0.001917z + 0.0004464}{z^3 - 2.722z^2 + 2.479z - 0.7524}$$

Je třeba si všimnout velkých rozdílů v časových konstantách jmenovatele a čitatele. Právě tento rozdíl způsobuje problémy při krácení kořenů v dalších fázích výpočtu. Pokud zaokrouhlíme velkou časovou konstantu na pět desetinných míst, na výsledek to nebude mít vliv, ale u nízké časové konstanty může být vliv značný.

Příklad: $T_1=5,123456$ po zaokrouhlení $T_1=5,12346$

$T_2=0,000004$ po zaokrouhlení $T_2=0$

Vzorce pro výpočet zpětnovazebního regulátoru podle kvadraticky optimálního kritéria:

$$N(q) = \frac{G.C^*.Pq + (1-q).A^-.R}{B^+} \quad M(q) = \frac{F.C^*.Pq - q^v.B^-.R}{A^+}$$

Zlomky musí být možno zkrátit tak, aby ve jmenovateli byl polynom 0-tého stupně.

Významy jednotlivých proměnných viz. Literatura [2],[12].

1. Volíme proměnné

```
nula=-12;      %s jakou presnosti urcujeme nulu
vysledek=-12;  %presnost zaokrouhleni vysledku
koreny=-2;     %presnost pri porovnani korenu
```

Výpočet regulátoru

```
[N,del]=kraceni(G*Ch*Pq+(1-q)*Aminus*R,Bplus);
[M,del]=kraceni(F*Ch*Pq-q^ny*Bminus*R,Aplus);
```

$$G^*Ch^*Pq+(1-q)^*Aminus^*R = -5.15 \times 10^{-4} + 0.0013q - 9.26 \times 10^{-4}q^2 + 6.88 \times 10^{-5}q^3 + 9.66 \times 10^{-5}q^4 - 7.00 \times 10^{-10}q^5 + 7.14 \times 10^{-10}q^6$$

$$F^*Ch^*Pq-q^{ny}^*Bminus^*R = -0.42 + 2.05q - 4.20q^2 + 4.63q^3 - 2.89q^4 + 0.97q^5 - 0.14q^6 + 1.73 \times 10^{-9}q^7$$

$$Bplus = 5.15 \times 10^{-4} + 1.28 \times 10^{-4}q \quad Aplus = 1 - 2.72q + 2.48q^2 - 0.75q^3$$

polynomy vstupující do krácení

```
>> roots(G*Ch*Pq+(1-q)*Aminus*R)
ans =
    1.0e+002 *
    0.00845722429677 + 3.67784715640182i
    0.00845722429677 - 3.67784715640182i
-0.04006310821626
    0.01276187585464
    0.01009250088945 + 0.00151326547044i
    0.01009250088945 - 0.00151326547044i

>> roots(Bplus)
ans =
-4.00656909285240
```

po vykrácení kořenů

```
>> roots(fliplr(N))
ans =
    1.0e+002 *
    0.00847053854526 + 3.67904676486105i
    0.00847053854526 - 3.67904676486105i
    0.01276187683251
    0.01009250062952 + 0.00151326531479i
    0.01009250062952 - 0.00151326531479i

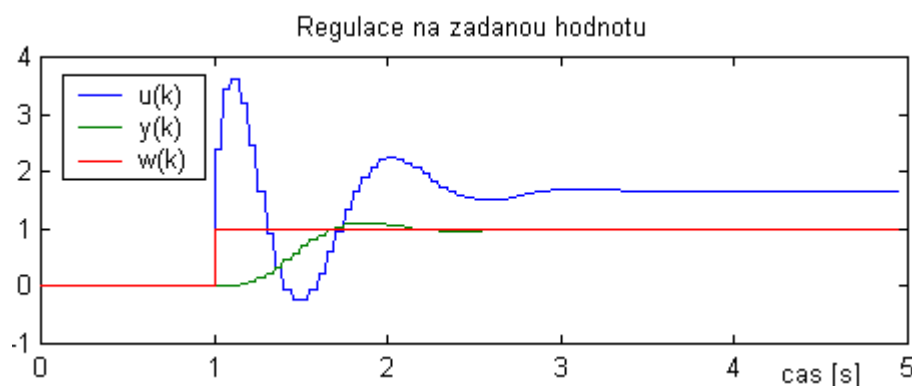
>> roots(fliplr(del))
ans =
Empty matrix: 0-by-1
```

Obr. 6.2.3-2 Krácení kořenů v programu Matlab (příklad 1)

Obdobně vypadá výsledek krácení při výpočtu polynomu M.

V tomto případě je vidět, že byly proměnné zvoleny dobře. Výsledkem je stabilní regulátor.

$$R_w(q) = \frac{2.39 - 6.5q + 5.92q^2 - 1.80q^3 + 0.000066q^4 - 0.000013q^5}{1 - 2.17q + 1.65q^2 - 0.43q^3 + 0.0000000055q^4}$$



Obr. 6.2.3-3 Průběh regulace na žádanou hodnotu (příklad 1)

2. Volíme proměnné

```
nula=-8;           %s jakou presnosti urcujeme nulu
vysledek=-12;      %presnost zaokrouhleni vysledku
koreny=-2;         %presnost pri porovnani korenu
```

Výsledkem je opět stabilní regulátor, ale stupně polynomů jsou menší.

$$R_w(q) = \frac{2.39 - 6.5q + 5.92q^2 - 1.80q^3}{1 - 2.17q + 1.65q^2 - 0.43q^3}$$

Tento rozdíl je způsoben ořezáním polynomů vstupujících do funkce kraceni. Změnila se hodnota proměnné nula a čísla, která jsou menší než 10^{-8} jsou považována za nulová.

$$\mathbf{G}^*\mathbf{Ch}^*\mathbf{Pq} + (1-q)^*\mathbf{Aminus}^*\mathbf{R} = -5.15 \times 10^{-4} + 0.0013q - 9.26 \times 10^{-4}q^2 + 6.88 \times 10^{-5}q^3 + 9.66 \times 10^{-5}q^4 - 7.00 \times 10^{-10}q^5 + 7.14 \times 10^{-10}q^6$$

$$\mathbf{F}^*\mathbf{Ch}^*\mathbf{Pq} - q^{\mathbf{ny}}^*\mathbf{Bminus}^*\mathbf{R} = -0.42 + 2.05q - 4.20q^2 + 4.63q^3 - 2.89q^4 + 0.97q^5 - 0.14q^6 + 1.73 \times 10^{-9}q^7$$

$$\mathbf{Bplus} = 5.15 \times 10^{-4} + 1.28 \times 10^{-4}q \quad \mathbf{Aplus} = 1 - 2.72q + 2.48q^2 - 0.75q^3$$

Výsledné průběhy jsou stejné jako u příkladu, kdy proměnná nula = -12. Je zde vidět, že vyšší stupně polynomů v regulátoru lze zanedbat, pokud jsou jejich hodnoty o několik řádů menší.

3. Volíme proměnné

```
nula=-8;           %s jakou presnosti urcujeme nulu
vysledek=-12;      %presnost zaokrouhleni vysledku
```

koreny=-5; %presnost pri porovnani korenu

Při porovnávání kořenů již nejsou nalezeny stejné, které by bylo možné vykrátit.

		Ar =	
		-4.00659688184951	
		1.27620249752198	
		1.00924871294264 + 0.15132421555305i	
		1.00924871294264 - 0.15132421555305i	
Hr =		Ar =	
-4.00656909285240		-4.00660000000000	
» roundn(Hr,-5)		1.27620000000000	
ans =		1.00925000000000 + 0.15132000000000i	
-4.00657000000000	≠	1.00925000000000 - 0.15132000000000i	

Obr 6.2.3-4 Krácení kořenů v programu Matlab (příklad 3)

V tomto případě není splněna podmínka krácení polynomů, a není tedy možné vypočítat regulátor.

Jestě horších výsledků můžeme dosáhnout v případě, že dojde k vykrácení stabilního a nestabilního kořene. Regulátor se vypočítá, ale nebude fungovat podle našeho očekávání.

Příklad: kořen polynomu A = 0,995

kořen polynomu B = 1,004

Při zaokrouhlení pomocí instrukce roundn(A,-2) popřípadě roundn(B,-2) jsou oba kořeny rovny jedné a budou vykráceny. Na výsledný regulátor může mít taková chyba fatální vliv.

6.2.4 zjistiqh.m

Při dělení dvou polynomů zjistí polynomy Q a H. $\frac{B}{A} = Q + \frac{H}{A}$

Je třeba rozlišovat tři možnosti vstupních dat a podle toho provést výpočet.

- 1) velikost polynomu A a B je stejná
- 2) velikost polynomu A je větší než velikost polynomu B
- 3) velikost polynomu A je menší než velikost polynomu B

V následujícím programu lze vidět všechny možnosti vstupních dat:

```
function [Qp,H] = zjistiqh(cit,jmen)      %cit=B jmen=A
gensym q
cit=pol2mat(cit);
jmen=pol2mat(jmen);
dc=length(cit);dj=length(jmen);
if dc==dj Qp=cit(1)/jmen(1);
    H=cit-Qp*jmen;
end;
```

} $\delta A = \delta B$

```

if dc<dj Qp=0;
    H=cit;
end;
if dc>dj
    roz=dc-dj;
    for i=1:roz+1
        Qp(i)=cit(1)/jmen(1);
        temp=-Qp(i)*mat2pol(jmen)*q^(roz+1-i);
        cit=temp+mat2pol(cit);
        cit=pol2mat(cit);
    end;
    H=cit;
end;
Qp=mat2pol(Qp);
H=mat2pol(H);

```

$\delta A > \delta B$

$\delta A < \delta B$

Příklad: 1) $\delta A = \delta B$

» A= 6+3*q+q^2
 » B= 1+2*q+3*q^2
 » [Q,H] = zjistiqh (B,A)

výsledek → Q = 3
 H = -17-7q

2) $\delta A > \delta B$

» A= 6+3*q+q^2+4*q^3
 » B= 1+2*q+3*q^2
 » [Q,H] = zjistiqh (B,A)

výsledek → Q = 0
 H = 1+2q+3q^2

3) $\delta A < \delta B$

» A= 6+3*q+q^2
 » B= 1+2*q+3*q^2+4*q^3
 » [Q,H] = zjistiqh (B,A)

výsledek → Q = -9+4q
 H = 55+5q

6.2.5 carka.m

Vytvoří adjungovaný polynom

$$\bar{A}(q) = A\left(\frac{1}{q}\right) = a_0 + a_1 \frac{1}{q} + \dots + a_n \frac{1}{q^n}$$

V programu Matlab není možné napsat polynom v záporných mocninách q.

Je tedy nutné předchozí vzorec upravit na tvar:

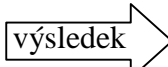
$$\bar{A}(q) = \frac{a_0 q^n + a_1 q^{n-1} + \dots + a_n}{q^n}$$

Čitatel odpovídá v programu proměnné Ac a jmenovatel proměnné del.

```

function [Ac,del] = carka(A)
dA=length(A);A=fliplr(A);
Ac=mat2pol(fliplr(A));
del=q^(dA-1);

```

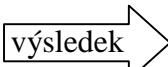
Příklad: $A(q) = (1-2q)(1-0,4q) = 1-2,4q+0,8q^2$
 $\gg A=[1 \ -2.4 \ 0.8]$
 $\gg [Ac,del]=carka(A)$  **výsledek** $Ac = 0.8 - 2.4q + q^2$
 $del = q^2$

6.2.6 hvezdicka.m

Vytvoří reflexní polynom $A^*(q) = A^+ \tilde{A}^-$

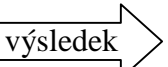
Je třeba dávat pozor na správné používání příkazů `pol2mat`, `mat2pol` v kombinaci s příkazem `fliplr`. Pokud máme polynom zadaný jako vektor $A=[1 \ 2 \ 3]$, to podle našeho předpokladu odpovídá polynomu $1+2q+3q^2$, pak při použití příkazu `mat2pol(A)` dostaneme reciprotý polynom $3+2q+q^2$. Pro správnou interpretaci polynomu je třeba zadat příkaz `mat2pol(fliplr(A))`.

```
function [Ah] = hvezdicka(A)
[Aplus,Aminus,qnany] = pm(A);
Ax=mat2pol(fliplr(pol2mat(Aminus)));
Ah=Aplus*Ax;
```

Příklad: $A(q) = (1-2q)(1-0,4q) = 1-2,4q+0,8q^2$
 $\gg A=[1 \ -2.4 \ 0.8]$
 $\gg [Ah]=hvezdicka(A)$  **výsledek** $Ah = -2+1.8q-0.4q^2$

6.2.7 oriznuti.m

Mějme polynom A,B. Pokud lze z obou vytknout mocninu q, pak je pokrátí. Tato funkce je obdobná jako funkce `kraceni`, ale v některých případech výpočtu regulátorů potřebujeme pokrátit pouze mocniny q a ne ostatní kořeny.

Příklad: $\gg A= q^2-q^3+2*q^4$
 $\gg B=2*q^3+q^4$  **výsledek** $B = 2q+q^2$
 $\gg [B,A]=oriznuti(B,A)$ $A = 1-q+2q^2$

V následující tabulce jsou všechny funkce toolboxu CR a formáty dat vstupujících a vystupujících z funkcí.

volání funkce	formát vstupních dat		formát výstupních dat	
[Aplus, Aminus ,ny] = pm (A)	A	$[q^0 q^1 \dots q^n]$	Aplus , Aminus	polynom
			ny	číslo (mocnina)
[Ao,Ho] = kraceni (A,H)	A , H	$[q^0 q^1 \dots q^n]$	Ao , Ho	$[q^0 q^1 \dots q^n]$
Ah = hvezdicka (A)	A	$[q^0 q^1 \dots q^n]$	Ah	polynom
[B,A] = oriznuti (B,A)	B , A	polynom	B , A	polynom
[Ac,del] = carka (A)	A	$[q^0 q^1 \dots q^n]$	Ac	polynom
			del	polynom ($q^a a_n$)
[Q,H] = zjistiqh (B,A)	B , A	polynom	Q , H	polynom

Tab. 6.2.-1 Přehled formátů dat vstupujících a vystupujících z funkcí toolboxu CR

7. Realizované úlohy

Nebudu zde popisovat jednotlivé programy pro výpočet regulátorů, ale uvedu pouze přehled programů a možností vstupních a výstupních dat. Vytvořil jsem dva druhy programů:

7.1 Funkce pro výpočty do Matlabu

Tyto funkce je možno použít jako standartní instrukce při práci v Matlabu. Je možné je nainstalovat obdobně jako toolbox CR (viz. kapitola 6.1 Instalace toolboxu CR).

Funkce by měly studentům pomoci pochopit programování těchto úloh v programu Matlab tak, aby byl každý student schopen vytvořit vlastní program.

- dopredny_1st_porucha.m
dopředný regulátor na poruchu v konečném počtu kroků
- zpetny_1st_porucha.m
zpětnovazební regulátor na poruchu v konečném počtu kroků
- zpetny_1st_w.m
zpětnovazební regulátor na žádanou hodnotu v konečném počtu kroků
- zpetny_2st.m
zpětnovazební regulátor se dvěma stupni volnosti v konečném počtu kroků
- dopredny_kv_1st_porucha.m
dopředný regulátor na poruchu podle kvadraticky optimálního kritéria
- zpetny_kv_1st_porucha.m
zpětnovazební regulátor na poruchu podle kvadraticky optimálního kritéria
- zpetny_kv_1st_w.m
zpětnovazební regulátor na žádanou hodnotu podle kvadraticky optimálního kritéria
- zpetny_kv_2st.m
zpětnovazební regulátor se dvěma stupni volnosti podle kvadraticky optimálního kritéria
- diofan.m
program pro výpočet diofantické rovnice, zejména při výpočtu časově optimálního diskrétního ovládání

volání funkce	formát vstupních dat		formát výstupních dat	
[M,N] = dopredny_1st_porucha (A,B,C,mocniny) [M,N] = dopredny_kv_1st_porucha (A,B,C,mocniny)	A,B,C	$[q^0 q^1 \dots q^n]$ nebo $[z^n z^{n-1} \dots z^0]$	M,N	$[q^0 q^1 \dots q^n]$
	mocniny	'z' nebo 'q'		
[M,N] = zpetny_1st_porucha(A,B,C,D,mocniny)	A,B,C,D	$[q^0 q^1 \dots q^n]$ nebo $[z^n z^{n-1} \dots z^0]$	M,N	$[q^0 q^1 \dots q^n]$
	mocniny	'z' nebo 'q'		
[M,N] = zpetny_1st_w(A,B,S,H,mocniny)	A,B,S,H	$[q^0 q^1 \dots q^n]$ nebo $[z^n z^{n-1} \dots z^0]$	M,N	$[q^0 q^1 \dots q^n]$
	mocniny	'z' nebo 'q'		
[M,N,P,X] = zpetny_2st(A,B,H,mocniny)	A,B,H	$[q^0 q^1 \dots q^n]$ nebo $[z^n z^{n-1} \dots z^0]$	M,N P,X	$[q^0 q^1 \dots q^n]$
	mocniny	'z' nebo 'q'		
[G,F,M,N,Y,Z] = zpetny_kv_1st_porucha (A,B,C,D,kappa,mocniny)	A,B,C,D	$[q^0 q^1 \dots q^n]$ nebo $[z^n z^{n-1} \dots z^0]$	G,F M,N Y,Z	$[q^0 q^1 \dots q^n]$
	mocniny	'z' nebo 'q'		
	kappa	číslo		
[G,F,M,N,Y,Z] = zpetny_kv_1st_porucha (A,B,S,H,kappa,mocniny)	A,B,S,H	$[q^0 q^1 \dots q^n]$ nebo $[z^n z^{n-1} \dots z^0]$	G,F M,N Y,Z	$[q^0 q^1 \dots q^n]$
	mocniny	'z' nebo 'q'		
	kappa	číslo		
[M,N,P] = zpetny_kv_2st(A,B,S,H,kappa,mocniny)	A,B,S,H	$[q^0 q^1 \dots q^n]$ nebo $[z^n z^{n-1} \dots z^0]$	M,N P	$[q^0 q^1 \dots q^n]$
	mocniny	'z' nebo 'q'		
	kappa	číslo		
[E,U]=diofan(B,A,S,H,z);	B,A	$[s^n s^{n-1} \dots s^0]$	E,U	polynomy v mocninách q
	S, H	$[q^0 q^1 \dots q^n]$ nebo $[z^n z^{n-1} \dots z^0]$		
	z	'z' nebo 'q'		

Tab. 7.1-1 Přehled formátů dat vstupujících a vystupujících z funkcí

V tabulce jsou proměnné pojmenovány stejně jako ve výpočtových programech (popřípadě v literatuře). Uživatel však může do funkce psát jiné názvy vstupujících a vystupujících proměnných, ale měl by si být vždy vědom toho, co která proměnná znamená.

7.2 Úlohy v Matlab Web Serveru

Tyto programy vykonávají stejný výpočet jako funkce do Matlabu, ale jsou rozšířeny o instrukce pro podporu Matlab Web Serveru (viz kapitola 5.3 Zpracování na straně serveru).

Klient, který přistupuje k těmto funkcím přes internetový prohlížeč, nemá přístup ke zdrojovým textům programu. Na stránkách předmětu je volně ke stažení verze funkcí pro Matlab, aby si studenti mohli prohlédnout i zdrojové kódy.

Názvy funkcí jsou odvozeny podobně jako v případě funkcí pro program Matlab, ale začínají znaky „cr_“. Ve stejném adresáři jsou i funkce pro podporu předmětu „Teorie automatického řízení“, které začínají znaky „tar_“

zpetny_kv_2st.m \longrightarrow cr_zpetny_kv_2st.m

Pro názornost popíšu zdrojový text programu pro výpočet dopředného regulátoru na poruchu, spočteného podle kvadraticky optimálního kritéria (viz. Literatura [2],[12]).

Základní vztahy:

$$Xq^s B^- + Y(1-q)A^* = C\tilde{B}^- \quad R_{opt} = \frac{-X}{B^*}$$

Příklad 1) v programu Matlab

Tento program voláme v Matlabu jako funkci. Deklaraci je vidět na prvním řádku programu. Je třeba dát pozor na správné názvy proměnných vstupujících a vystupujících z funkce a na jejich typ.

```
function [M,N] = dopredny_kv_1st_porucha(A,B,C,mocniny)
if mocniny == 'z' %výpočet je pro polynomy v mocninách q, pokud nejsou pak převedeme
    A=fliplr(A);B=fliplr(B);C=fliplr(C);
end
[Aplus,Aminus,ksi] = pm(A);
[Bplus,Bminus,ny] = pm(B);
[Cplus,Cminus,my] = pm(C);
qs=q^(ny-my+(length(fliplr(pol2mat(Bminus))))-1)) %vypočte q na s
Ah=hvezdicka(A);Bh=hvezdicka(B); %vytvoříme reflexní polynom z A a B
Bv=mat2pol(fliplr(pol2mat(Bminus))); %vytvoří reciprotý polynom z B-

[X,Y]=axbyc(qs*Bminus,(1-q)*Ah,Cplus*Cminus*Bv,'minx'); %výpočet lineární diofantické rovnice
M =fliplr (pol2mat(Bh)); %jmenovatel optimálního regulátoru
N =fliplr (pol2mat(-X)); %čitatel optimálního regulátoru
```

Příklad 2) v programu Matlab Web Server

Výpočtová část programu bude stejná jako v příkladě 1, ale musí být doplněna o načítání vstupních dat, zápis výstupních dat, simulaci průběhů. Všechny tyto kroky jsou popsány v kapitole 5.3 Zpracování na straně serveru (m-file).

8. Odkazy na hypertextové dokumenty

Hlavní stránka předmětu „Číslicové řízení“

http://www.fm.vslib.cz/~krtsub/fm/cir/krt_cir8.htm

Studijní materiály pro podporu výuky předmětu „Číslicové řízení“

<http://www.fm.vslib.cz/~krtsub/fm/cir/materialy.htm>

Modrlák O., Analýza diskrétních regulačních obvodů

http://www.fm.vslib.cz/~krtsub/fm/cir/skripta/cr_analyza.pdf

Modrlák O., Modelování a diskrétní identifikace

http://www.fm.vslib.cz/~krtsub/fm/cir/skripta/cr_stochas.pdf

Modrlák O., Syntéza číslicového regulátoru

http://www.fm.vslib.cz/~krtsub/fm/cir/skripta/cr_synteza.pdf

Modrlák O., Syntéza diskrétních stavových regulátorů

http://www.fm.vslib.cz/~krtsub/fm/cir/skripta/cr_stavovy.pdf

Popis funkcí toolboxu CR

http://www.fm.vslib.cz/~krtsub/fm/cir/manualy/toolbox_CR.pdf

Nejpoužívanější funkce v Polynomial toolboxu

http://www.fm.vslib.cz/~krtsub/fm/cir/manualy/toolbox_polyx.pdf

Stručný slovník používaných výrazů

<http://www.fm.vslib.cz/~krtsub/fm/cir/skripta/slovnicek.pdf>

Toolbox CR (funkce do programu Matlab)

http://www.fm.vslib.cz/~krtsub/fm/cir/programy/cr_toolbox.rar

Výpočty regulátorů (funkce do programu Matlab)

http://www.fm.vslib.cz/~krtsub/fm/cir/programy/cr_regulatory.rar

Časově optimální diskrétní ovládání (funkce do programu Matlab)

http://www.fm.vslib.cz/~krtsub/fm/cir/programy/cr_diofan.rar

Úlohy pro Matlab Web Serveru (MWS)

<http://147.230.128.46/uvod.html>

Dopředný regulátor na poruchu v konečném počtu kroků

http://147.230.128.46/fce/cr_dopredny_1st_porucha_in.html

Zpětnovazební regulátor na poruchu v konečném počtu kroků

http://147.230.128.46/fce/cr_zpetny_1st_porucha_in.html

Zpětnovazební regulátor na žádanou hodnotu v konečném počtu kroků

http://147.230.128.46/fce/cr_zpetny_1st_w_in.html

Zpětnovazební regulátor 2 stupně volnosti v konečném počtu kroků

http://147.230.128.46/fce/cr_zpetny_2st_in.html

Dopředný regulátor na poruchu podle kvadraticky optimálního kritéria

http://147.230.128.46/fce/cr_dopredny_kv_1st_porucha_in.html

Zpětnovazební regulátor na poruchu podle kvadr. optimálního kritéria

http://147.230.128.46/fce/cr_zpetny_kv_1st_porucha_in.html

Zpětnovazební regulátor na žádanou hodnotu podle kvadr. optim. kritéria

http://147.230.128.46/fce/cr_zpetny_kv_1st_w_in.html

Zpětnovazební regulátor 2 stupně volnosti podle kvadr. optimálního kritéria

http://147.230.128.46/fce/cr_zpetny_kv_2st_in.html

Časově optimální diskrétní ovládání

http://147.230.128.46/fce/cr_cas_opt_ovl_in.html

Internetové stránky používané při tvorbě diplomové práce

Výzkumná skupina DSP - Sborníky Matlab konferencí

<http://phobos.vscht.cz/CZECH/matlab.asp>

Výzkumná skupina DSP - Aplikace Matlab web serveru:

<http://phobos.vscht.cz/CZECH/amlweb.asp>

Matlab web server - ukázky

<http://hpk.felk.cvut.cz/~xnovakd1/mws/mws.html>

Diskuzní forum o Matlabu

<http://forum.builder.cz/list.php?f=141>

The Mathworks - Matlab web server

<http://www.mathworks.com/products/webserver/index.shtml>

Vysoká škola Báňská – Teorie řízení

<http://www.fs.vsb.cz/books/MatMet/ATR.htm>

Problémy při tvorbě internetových stránek

<http://www.w3.org/Arena/tour/symbols.html>

PolyX, Ltd

<http://www.polyx.cz>

Golden HTML editor

<http://golden.zde.cz/>

Závěr

V rámci diplomové práce byla vytvořena internetová podpora předmětu „Číslicové řízení“. Byla zpracována úroveň informační, autokontrolní a interaktivní.

- 1) *Informační část:* Vytvořil jsem internetové stránky obsahující elektronická skripta, která pokrývají celý obsah přednášené látky, a manuály popisující toolboxy, využívané při programování v Matlabu. Je zde možno nalézt rozvrhy přednášek a cvičení pro jednotlivé týdny výuky i zadání semestrálních prací.
- 2) *Autokontrolní část:* Pro podporu teoretických znalostí studentů jsem vytvořil několik kvízů s otázkami, které pokrývají přednášenou látku a jsou nezbytné k pochopení dalších souvislostí.
- 3) *Interaktivní část:* Na tuto část byl kladen největší důraz při tvorbě diplomové práce. Matlab Web Server byl rozšířen o úlohy pro číslicovou syntézu s možností změny parametrů. Na internetových stránkách je možno nalézt programy pro výpočet různých druhů regulátorů, které jsou realizované pomocí Matlab Web Serveru a může je tedy spouštět i uživatel, který nemá nainstalovaný program Matlab. Dále byl vytvořen toolbox CR, který by měl zjednodušit studentům programování úloh díky realizaci nejčastěji používaných rutin. Při programování jednotlivých funkcí bylo přihlédnuto k numerickým chybám, které se objevují při řešení na reálných celcích, a jejich použití je tedy možné i pro praktická řešení regulace.

Do budoucna by bylo vhodné vyřešit problémy vzniklé při programování toolboxu CR, jako je například krácení polynomů (viz. kapitola 6.2.3 *kraceni.m*). Tento problém je možné vyřešit pomocí dynamické změny parametrů, která by však několikanásobně zpomalila výpočet.

Doba výpočtu je velmi důležitý faktor a jelikož program Matlab není optimalizován z hlediska výpočetní rychlosti, bylo by vhodné se zamyslet, zda by se úlohy nedaly realizovat v jiném programovacím jazyce i za cenu složitějšího programování základních operací.

Literatura

- [1] KYRAL, A. *Hypertextový studijní materiál předmětů teorie řízení Diplomová práce*. Liberec: TU Liberec, 2002. 62.
- [2] MODRLÁK, O. *Teorie automatického řízení (cvičení)*. Liberec: Ediční středisko VŠST, 1992. 270.
- [3] KUČERA, M. *HTML – tipy a triky od profesionálů*. Brno: UNIS Publishing, s.r.o., 2001. 80. ISBN 80-86097-64-1
- [4] KRISTIÁN P. *Návrh webu a FrontPage 2000*. Brno: UNIS Publishing, s.r.o., 2001. 79. ISBN 80-86097-57-9
- [5] BROŽA P. *Programování WWW stránek pro úplné začátečníky*. Praha: Computer Press, 2000. 172. ISBN 80-7226-278-5
- [6] ROUBÍČEK L. *Microsoft Windows 2000 Server – Správa systému*. Praha: Computer Press, 2000. 600. ISBN 80-7226-291-2
- [7] HLAVENKA J. *Výkladový slovník výpočetní techniky a komunikací*. Praha: Computer Press, 1997. 456. ISBN 80-7226-023-5
- [8] SCOTT I. *Dynamické HTML*. Praha: Computer Press, 2000. ISBN 80-7226-268-8
- [9] KVĚTOŇ K. *Základy online výuky a eLearning: BELCOM 02*, ČVUT Praha, 2002. dostupné na <http://web.cvut.cz/online>
- [10] MODRLÁK, O. *Analýza diskrétních regulačních obvodů - Studijní materiál*. Liberec: TU Liberec, 2004. dostupné na http://www.fm.vslib.cz/~krsub/fm/cir/skripta/cr_analyza.pdf
- [11] MODRLÁK, O. *Modelování a diskrétní identifikace - Studijní materiál*, Liberec: TU Liberec, 2004. dostupné na http://www.fm.vslib.cz/~krsub/fm/cir/skripta/cr_stochas.pdf
- [12] MODRLÁK, O. *Syntéza číslicového regulátoru - Studijní materiál*, Liberec: TU Liberec, 2004. dostupné na http://www.fm.vslib.cz/~krsub/fm/cir/skripta/cr_synteza.pdf
- [13] MODRLÁK, O. *Základy číslicového řízení - Studijní materiál*, Liberec: TU Liberec, 2003. dostupné na http://www.fm.vslib.cz/~krsub/fm/tr2/tar2_zcr.pdf

Přílohy

Příklady ze studijních materiálů:

- | | | |
|--|--|---|
| <p> tar2_zas</p> <ul style="list-style-type: none"> • Příklad 1.3.1, str. 12 • Příklad 1.3.2, str. 13 • Příklad 1.3.3, str. 16 • Příklad 1.9, str. 24 • Příklad 2.1, str. 27 • Příklad 2.2, str. 32 • Příklad 3.1, str. 39 • Příklad 3.2, str. 43 | <p> tar2_nel</p> <ul style="list-style-type: none"> • Příklad 2.5.2, str. 28 • Příklad 2.5.3, str. 29 | <p> Číslicové řízení</p> <ul style="list-style-type: none"> • Dopředný regulátor na poruchu • Zpětnovazební regulátor na poruchu • Zpětnovazební regulátor na žádanou hodnotu • Zpětnovazební regulátor 2 stupně volnosti • Dopředný regulátor na poruchu (kvadratický) • Zpětnovazební regulátor na poruchu (kvadratický) • Zpětnovazební regulátor na žádanou hodnotu (kvadratický) • Zpětnovazební regulátor 2 stupně volnosti (kvadratický) • Časově optimální diskrétní ovládání |
|--|--|---|

Odkazy:

- [Materiály pro předmět AR1](#)
- [Materiály pro předmět AR2](#)
- [Materiály pro předmět CR](#)
- [Manuál Control WEBu 2000](#)

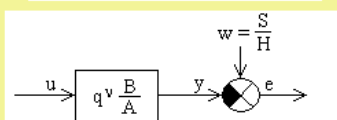


Fakulta Mechatroniky a mezipředmětových inženýrských studií
Technická univerzita v Liberci - Katedra řídicí techniky
Autor: Aleš Kyrál a Pavel Šír

Obr. P-1 Seznam úloh realizovaných pomocí Matlab Web Serveru

CR - Časově optimální diskrétní ovládání - není použita aktuální informace z výstupu systému

$$EH + q^{-1}BU_1 = S \quad U_1 = U \frac{H}{A}$$



Hledáme takové ovládání u , aby odchylka e vyhovovala silné verzi konečného počtu kroků řízení.

V jakých mocninách zadáváme (q nebo z):

Přenosy:

A: B:

S: H:

Obecné:

vzorkovací čas: doba simulace:

Které grafy chcete zobrazit:

u: ☒ y: ☒ e: ☐ w: ☒

Obr. P-2 Úloha pro výpočet časově optimálního diskrétního ovládání

Studijní materiály:

1. Tématicky zpracované okruhy

- [Analýza diskrétních regulačních obvodů](#)
- [Modelování a diskrétní identifikace](#) - [1.kviz](#) [2.kviz](#) [3.kviz](#)
- [Syntéza číslicového regulátoru](#)
- [Příloha - Z-Transformace](#)
- [Stručný slovník](#), aneb co by měl znát každý.

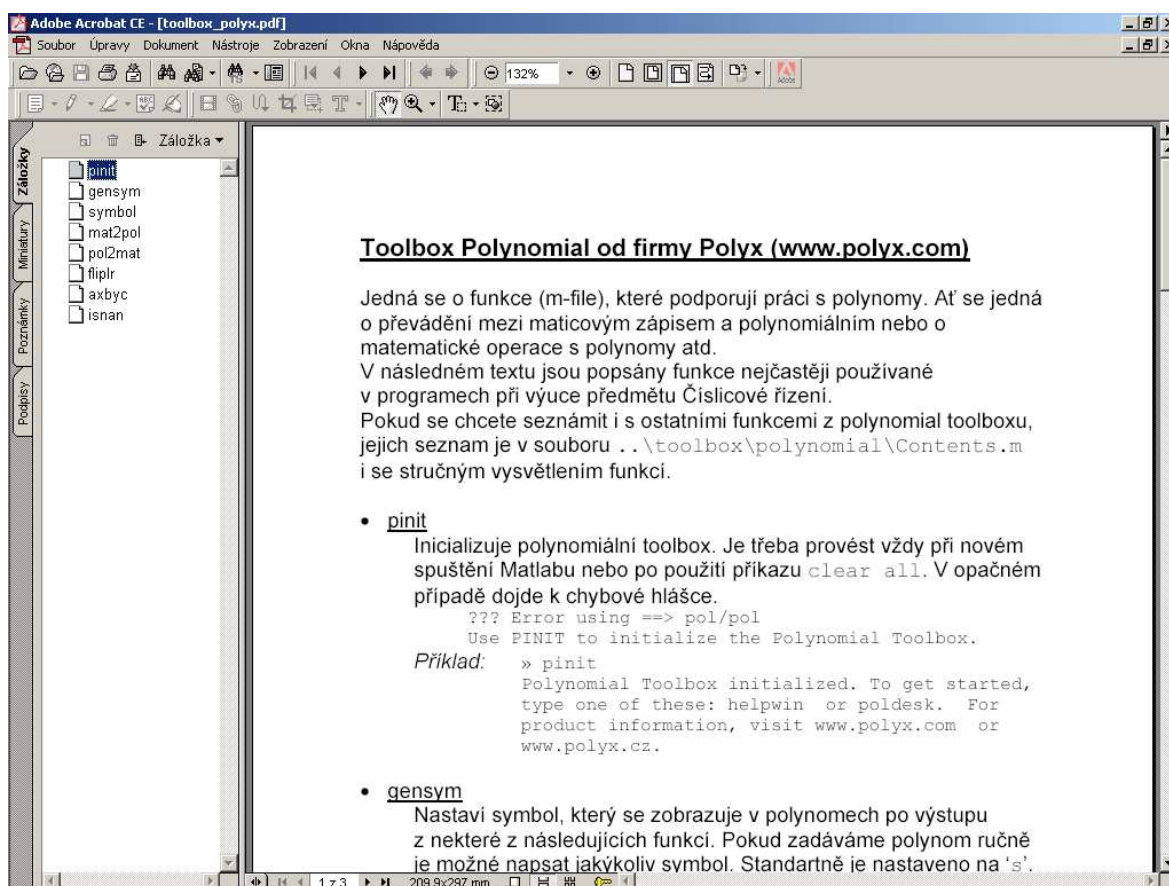
2. Dokumenty k programové podpoře předmětu.

- [Stručný manuál MATLABu pro předměty Teorie Řízení](#)
- [Manuál k toolboxu CR](#)
- [Stručný manuál k Polynomial toolboxu](#) - [Kompletní přehled instrukcí Polynomial toolboxu](#)

3. Programová podpora předmětu.

- [MATLAB WEB SERVER](#) - podpora předmětů TAR1, TAR2, CR
- [Toolbox CR](#) - nejčastější funkce používané v předmětu Číslicové Řízení (rar - 2897 B)
- [Funkce do Matlabu pro výpočty regulátorů v předmětu Číslicové Řízení](#) (rar - 4293 B)
- [Funkce do Matlabu pro výpočet časově optimálního diskrétního ovládání](#) (rar - 1013 B)

Obr. P-3 Studijní materiály předmětu Číslicové řízení



Obr. P-4 Popis některých funkcí z polynomial toolboxu